ISSUE 20 OCTOBER 2023 - WWW.RETROMAGAZINE.NET - FREE PUBLISHING

# RetroMagazine World

future days are back

ISSN 3103-697X

## A PIG QUEST

COMMODORE 64

INTERVIEW WITH
KAZUHIKO NISHI

HARDWARE
ATARI 5200 SuperSystem

CALC RESULT: The Commodore 64 Spreadsheet
PROGRAMMING: An Experiment in Assembly for the C64
MSX Basic optimization and Assembly support
Graphics... What a passion! Part IV - The KLA graphic format
Nintendo DS: The (UN)protections
... columns, interviews, reviews and much more!

## There is still so much to be done and we need you!

These days I found myself talking to a person who kept and preserved some of his gaming companions from the 80s; specifically a Commodore 64 and a Commodore 16.
As we continued our conversation, I realized how he would enjoy sharing the experience of playing with these machines with his son.

So far nothing out of the ordinary, we have seen dozens of photos of retro-computing enthusiasts proudly showing their children playing amusedly with the computers and consoles that cheered our childhood.

What amazed me, instead, was the fact that he had no idea that these machines are nowadays more alive than ever. Given the proliferation of websites, social profiles, paper and online magazines dedicated to retro-computing, and even the interest of the press and generalist TV, I would have expected that the topic would not be so completely obscure. Probably our being part of the system makes us take many things for granted, but that is unfortunately not the case. There are still thousands of people who have no idea that they could somehow relive their youth...
And it is a pity that they are missing this opportunity.

Clearly there is still so much to be done.
We need to be able to reach as many people as possible.
And to do this we need your help.

RetroMagazine World has been planning for some time now to put online all the articles that have been published in the pages of the magazine. This would increase the visibility of our work and consequently we could reach out to more and more people.
Okay, but what does that have to do with us, you may ask. Unfortunately, our time is limited and what we have left over, we already devote to the production of the articles, the magazine and all the other RetroMagazine World projects. So we are looking for a few volunteers who would like to be part of our team by helping us with the transposition of the articles to the site.

If the idea of collaborating with us by helping us spread the word about retro-computing appeals to you, please feel free to contact us through our official channels!

We do not promise you money and fame, but the eternal gratitude of thousands of fans. Maybe... :-D

**Francesco Fiorentini**

**TABLE OF CONTENTS**


| | |
|---|---|
| ◊ Nintendo DS - the (un)protections | Pag. 3 |
| ◊ Atari 5200 SuperSystem | Pag. 8 |
| ◊ CALC RESULT -the Spreadsheet for the C64 | Pag. 14 |
| ◊ An assembly experiment for the C64 | Pag. 18 |
| ◊ Graphics... What a passion! Part IV - The KLA format and how to deal with it | Pag. 26 |
| ◊ Optimizing MSX BASIC and how to support assembly | Pag. 31 |
| ◊ A couple of words with the genius of Kobe | Pag. 34 |
| ◊ ListAmiga - Amiga's best 10 platform games | Pag. 37 |
| ◊ Neonnoir (Amiga) | Pag. 43 |
| ◊ Lester (C64) | Pag. 44 |
| ◊ Life of Mars (MD) | Pag. 45 |
| ◊ Bufonada (MSX/2 - Spectrum) | Pag. 46 |
| ◊ Mad Stalker Full Metal Forth (MD) | Pag. 48 |
| ◊ A Pig Quest (C64) | Pag. 50 |
| ◊ Aquabyss (Amiga) | Pag. 52 |
| ◊ Maria Renarda's Revenge (Amiga) | Pag. 54 |
| ◊ S.A.K. (MSX2) | Pag. 56 |
| ◊ Mystic Pillars (NES) | Pag. 57 |
| ◊ The Firemen (SNes) | Pag. 58 |
| ◊ Minky (Amiga) | Pag. 60 |
| ◊ Shadow Warrios Ninja Gaiden (GameBoy) | Pag. 61 |
| ◊ Crystalis (NES/GBC) | Pag. 63 |
| ◊ James Bond 007 - Live and Let Die (C64) | Pag. 64 |


**People involved in the preparation of this issue of RetroMagazine World** (in no particualr order):

- Dr. Andrea Q.
- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- Francesco Fiorentini
- Leonardo Miliani
- Roberto Del Mar Pirazzini
- Ingrid Poggiali
- Giampaolo Moraschi
- Giuseppe Rinella
- Eugenio Rapella
- Germán Gómez Herrera
- Takahiro Yoshioka
- Barbara "Morgana" Murgida
- Cover image: **Giuseppe Mangini**
- Cover design: **Carlo N. Del Mar Pirazzini**

# Nintendo DS - the (un)protections

*by Dr. Andrea Q. - www.retrofixer.it*

*Youtube channel: https://www.youtube.com/channel/UCEw0CQ8LKyA9jVvWXkEwp4Q*

And here we are at a new installment of this apparently popular column.

After covering the now-no-longer-mysterious iQue Player, today we try to unveil what lies inside Nintendo's next handheld console.

Released at the end of 2004 and "upgraded" later with the "Lite" version in 2006, the Nintendo DS began to have several features that we would later find in future consoles from this major video game and consoles company.

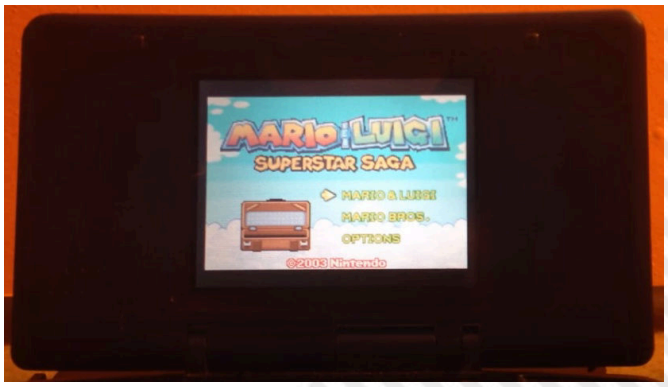
On the right of the photo you can see the "Fat" version while on the left is the "skinny" Lite version.

The main differences between the 2 versions are as follows. The DS Lite is:
- about 42% smaller and 21% lighter while retaining all the features of the "normal" DS
- its backlight is always on and has 4 adjustable brightness levels
- aesthetically, the case is brighter, the speakers take up less space, and the position of buttons and microphone has been changed
- the touch stylus got bigger
- because of the smaller size, the GBA cartridges protrude from the Lite system; therefore, in order to give a feeling of continuity of profiles, a cartridge is included to act as a "filler" when there are no GBA games inserted in the appropriate slot.

Its product code is NTR-001 (USG-001 for the Lite version) and is a contraction of its development code name "Nitro"; the meaning of the suffix DS stands for "Developers System" but also for "Dual Screen."

As Nintendo has accustomed us, this console is also backward compatible with games from its portable predecessor, the GBA, and booting can be done either from an NDS cartridge, a GBA cartridge, Firmware, or via WiFi. When a GBA cartridge (32pin) is inserted, the NDS automatically swtcha into GBA Mode and becomes "almost" like a real GBA.

The differences with a real GBA are as follows:
- Does not support GB/GBC cartridges
- The undocumented register Internal Memory Control (Port 800h) is not supported so NDS does not allow overclocked RAM
- there is no link port (thus no support for hardware link port) consequently GBA games can only be used in 1 player mode
- the audio frequency seems to go from 16.78MHz to 16.76MHz, so slightly slower than the original GBA (imperceptible difference but apparently documented)
- In the BIOS GBA changes a byte from the value 00 to 01 causing the checksum of the same to change
- The GBA logo can be made to appear in either the upper or lower screen (selectable option in the start menu)
- The GBA game screen has black bands on the side as in the picture below:

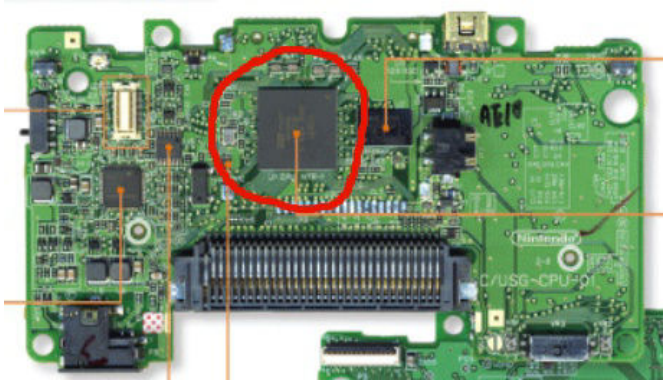



The engine of the DS is an ARM9 processor that cooperates with an ARM7 probably for reasons related to easier backward compatibility with the GBA, which is precisely equipped with an ARM7.

## THE PROTECTIONS

### BIOS

The BIOS of the ARM7 (called NDS7), about 45KB in size, is divided into 2 regions: the first one that is excluded from being read by writing the 1024h register just before reading the game cartridge as for the GBA and another one that is always active.



The protected part contains the cartridge KEY1 encryption seed used to encrypt/decrypt data stored on cartridge. In addition, the BIOS own functions have software protection that prevents them from reading the protected area.

The system to read the protected data is similar to that used for GBA and that is to use hardware tricks to read the memory before it is protected; alternatively, 2 opcodes (THUMB opcodes 05ECh and 05EEh) capable of bypassing this constraint were discovered.

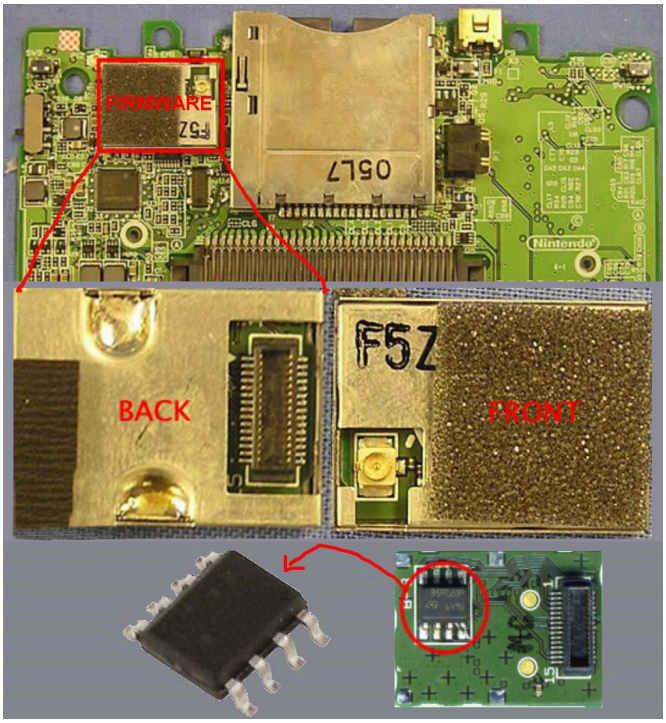
The BIOS of the ARM9 (NDS9), about 65KB in size, is not protected from being read by either hardware or software.

The BIOS checks the cartridge header (header ranging from offset 0x0000 to 0x0200), in particular it checks the Nintendo logo checksum stored at offset 0x15C (checksum calculated with CRC-16 algortim from offset 0x00C0 to offset 0x015B) and corresponds to a fixed value = CF56
The firmware later will also check the bytes of the logo itself (the Nintendo logo goes from offset 0x00C0 to offset 0x015B).

### THE FIRMWARE

The firmware is stored inside an NVRAM chip that has different sizes depending on the type of console and bundle:
ST M45PE20 - 256 KBytes (Nintendo DS)
ST M35PE20 - 256 KBytes (Nintendo DS-Lite)
ST M25PE40 - 512 KBytes (iQue DS, Chinese conc aracteries)
ST 45PE40V6 - 512 KBytes (DS Zelda, NTR-AZEP-0)
ST 45PE80V6 - 1024 Kbytes (e.g., Spirit Tracks, NTR-BKIP)
Sanyo LE25FW203T - 256 KBytes (Mariokart backup)

In each case the "core" is 256KB and is divided into 5 encrypted and compressed sections:
part1 - ARM9 BOOT CODE
part2 - ARM7 BOOT CODE
part3 - ARM9 GUI CODE
part4 - ARM7 GUI CODE
part5 - DATA/GFX
which are verified with checksums. The coder Chishm was able to reverse engineer the compression/encryption and developed a tool to extract the various parts of the firmware; the tool is called Fwunpacker. There is also a tool called CFW Suite to process a firmware and modify its GUI.




The ability to act at such a low level in the system also allowed the creation of a virus that could brick the console, apparently functioning only if you had done the softmod, showing a brick wall in the dual screen at startup:

It is, however, possible to do a "hot swap" of the chip containing the firmware, thus allowing you to recover brikked consoles by flashing a valid firmware with FlashMe.

## PROPRIETARY FORMAT CARTRIDGES

The cartridges used by the DS have a proprietary format with 17 PINs and have product code NTR-005; there is a variant with infrared hardware support with product code NTR-031:

As with the GBA, clones (real on the right, fake on the left) also exist for these cartridges:

For more details on how to recognize them go here.

## CRIPTED DATA

Data on cartridge can be encrypted.

Specifically, in the 0x4000-0x7FFF range of a cartridge there is a "SECURE AREA" accessible only to the ARM9 and decryptable by it through encrypted commands using the KEY1 seed; not all cartridges have the secure area; once decrypted, only the first 8 bytes (which contain the Secure Area ID) of this range are checked; if the check is correct the cartridge loading continues.

Games loaded via WLAN do not have the secure area and games that do have it cannot be started via WLAN.

The encryption algorithm is based on the "Blowfish Encryption Algorithm" devised by Bruce Schneier in 1993. However, cartridges are dumpable through specific SPI protocol commands; the dumpable chips of a cartridge are as follows:

Type / Total Size / Page Size / Chip-Example / Game-Example.

EEPROM 0.5K bytes 16 bytes ST M95040-W (eg. Metroid Demo)

EEPROM 8K bytes 32 bytes ST M95640-W (eg. Super Mario DS)

EEPROM 64K bytes 128 bytes ST M95512-W (eg. Downhill Jam)

FLASH 256K bytes 256 bytes ST M45PE20 (eg. Skateland)

FLASH 256K bytes Sanyo LE25FW203T (eg. Mariokart)

FLASH 512K bytes 256 bytes ST M25PE40? (eg. which/any games?).

FLASH 512K bytes ST 45PE40V6 (eg. DS Zelda, NTR-AZEP-0)

FLASH 1024K bytes ST 45PE80V6 (eg. Spirit Tracks, NTR-BKIP)

FLASH 8192K bytes MX25L6445EZNI-10G (Art Academy only, TWL-VAAV)

FRAM 8K bytes No limit ? (eg. which/any games?)

FRAM 32K bytes No limit Ramtr

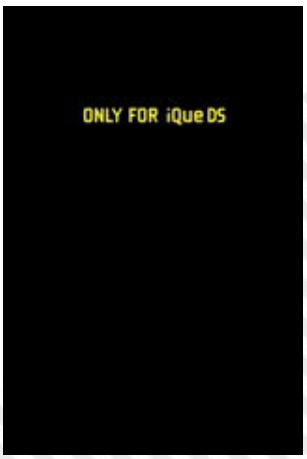
Note: FRAMs are RAM memories that hold data permanently without needing a battery (as opposed to SRAMs) and are faster than EEPROMs, with which they are still backward compatible.

Once recognition has taken place via KEY1-encrypted commands, data is exchanged through another algorithm called KEY2 based on 2 registers of 39 bits, of XORs and shifts. Note that ALL dumps of NDS cartridges found online have the header decrypted.

## REGION CHECK

The console has no region check.

Instead, iQue DS game cartridges contain a feature that checks the console's region and if they do not find it, they will not start the game (such protection is therefore in the game, not the console).




To remedy the problem, in DS consoles, the offsed 0x1D of the ROM must be edited and the value 80 changed to 00:

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | 4E | 49 | 4E | 54 | 45 | 4E | 44 | 4F | 47 | 53 | 00 | 00 | 41 | 33 | 36 | 43 | NINTENDOGS..A36C |
| 00000010 | 30 | 31 | 00 | 00 | 08 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01.............. |
| 00000020 | 00 | 40 | 00 | 00 | 00 | 08 | 00 | 02 | 00 | 00 | 00 | 02 | D8 | C0 | 0B | 00 | .@..........ØÀ.. |

## SYSTEMS FOR EXECUTING ARBITRARY CODE

### FIRMWARE V1 to V3

Before the encryption algorithm was cracked in 2006, various systems existed to boot unencrypted code that took advantage of several missed checks by the console firmware:

### PassMe

It is a hardware that interposes itself between the console and the cartridge in the GBA slot (SLOT-2), reads the ID from the latter tricking the system into thinking that the inserted flash cartridge is original. It was the basis for all software modifications to the console.

### WiFiMe

The DS has the ability to connect via WiFi and even send games via WiFi!

In fact, the DS Download Play service in the V1 firmware allows applications to be sent, and a flaw in the management of the original RSA signature of that system allowed hackers to create WiFiMe, a program to boot homebrew. In detail, the first version of the Download Play title uses an execution address from a header that is not part of the code and data digital signature calculation. WiFiMe was extracted from Super Mario 64 DS with the modified execution address in the GBA ROM space, but it could have been any DS Download Play program.

### Wireless Multiboot (WMB)

This Windows application created by coder FireFly requires d a PCI or PCMCIA network card with Ralink chipset; it can send small Nintendo-signed programs to the DS Download Play.

### FlashMe

This system overwrites the NDS firmware allowing immediate execution of DS code even in the GBA ROM space, removing the digital signature check for DS Download Play and thus allowing homebrew to be sent. We are looking at one of the very first softmods for consoles.

### FIRMWARE V4

Since firmware version V4, the bugs that allowed the aforementioned exploits have been patched but this has not deterred hackers who have gone ahead in the classic cat-and-mouse chase.

How to check the firmware version of the NDS:

We rely on the behavior of the PictoChart application: you start the DS with a game on, start PictoChart, and pull out the cartridge with the console still on; a color of the screens will appear that is different for each firmware version (this technique is also called "Pink Screen of Death" or "TakeMeOut"):

v1: Pictochat hangs/no color is shown

v2: two blue-gray screens

v3/iQue: two dark green screens

v4: two golden yellow screens

v5: two magenta screens (DS lite come out with this version, along with some pink DSs):

v6: two dark blue screens

v7: the system does not crash (only in the gold version of the Japanese DS Lite limited edition).

### PassMe2

Like PassMe but exploiting a different DS BIOS bug by using specific games (a list of supported ones is available here)

### FlashMe v5

In this version the program code is scaled down and written in protected areas of the firmware that cannot be rewritten by any original game. Previous versions of FlashMe were in fact written in areas that were supposed to be "unused" discovering later that this was not the case.

**NoPass**

From the moment in 2006 when the cartridge encryption algorithm was reversed by Martin Korth (author of the No$GBA emulator as well as the one who also first managed to dump the NDS bios) it was possible to execute code even from cartridges inserted in SLOT-1 (the one for NDS games); this method was named "NoPass."

**NoPass Cartridges**

Thus cartridges began to come out that did not require an original game to run arbitrary code because they were coded just like the original cartridges such as Datel's Max Media Launcher, which you can see in the photo below in comparison with a more dated PassMe:

Next came the dozens and dozens of R4s, M3s, AceKards, etc. each with their own boot firmwares some of them even reversed like the R4 one again thanks to Chishm and its r4crypt tool (there is also injektor2 1.08 - latest version 1.09 apparently unobtainable).

**SOUNDHAX**

In late 2014, the reverser/dev/coder Smealum showed a video of the audio signal-based hack for the game Bangai-o-Spirit; the game allows levels to be transferred via audio: by reversing the code, it was possible to buffer overrun and execute arbitrary code simply by sending a .WAV to play (you can also see it at work in the DeSmuME emulator).

**CONCLUSIONS**

As you can see things are starting to get very interesting in the world of console hacking especially because for the first time we are starting to have the possibility to arbitrarily write into the internal memory of a console so as to have a sort of very first and rudimentary custom firmware. We also saw how countermeasures were put in place for the first time to limit the damage caused by the bugs uncovered by the reversers: the nowadays dreaded FIRMWARE UPDATES!

Note that the console firmware could not update itself either via WiFi or via cartridge so when buying the console we went a little "by luck" when the "Pink Screen" method explained above was not yet known. Although very late there are also exploits that exploit flaws in games such as the SoundHax exploit.

I apologize if I may have seemed hasty on some topics but the DS was a rather unfamiliar console to me and I had to study it practically from scratch. If you have any questions or clarifications please write to me.

The next installment will discuss the evolution of this Nintendo in terms of both hardware and security.

## WARNING: Disclaimer

The information contained in this article is for informational purposes only. This documentation is not guaranteed to be error-free. If this information is used to modify your hardware, it is your responsibility to take all necessary emergency, backup, redundancy, and other measures to ensure its safe use. RetroMagazine World disclaims all liability for any damages caused by the use of the information in this article.

# Atari 5200 SuperSystem

*by Leonardo Miliani*

In 1982 Atari, then the undisputed ruler of the home video game market, unveiled the heir to the 2600/VCS, that console that had been literally making waves among electronic gaming systems for several years, present in millions of homes. Marketed as a direct competitor to the Mattel Intellivision, the new console would not only fail to hold its own against the latter but would also give way to the ColecoVision, which would arrive shortly before its debut, and would eventually be taken off the market even before the console whose place it was supposed to take. We are talking about the ill-fated Atari 5200 SuperSystem, which would have deserved better luck.

**Origins**

In 1975 Atari decided to develop an electronic device that could play its arcade games on an ordinary television set. Unlike Pong, which is a device based on discrete components and TTL integrateds, and thus can only run the game for which it was designed, Atari wants the new device to be "programmable," that is, one can easily change the game to be played. In the middle of that year MOS Technology unveiled the 6502 CPU, an inexpensive microprocessor that paved the way for the console's realization: in fact, Atari negotiated for the supply of a depowered version of the chip and put the VCS, from "Video Computer System," on the market. The console also owes much of its fortunes to a particular chip that flanks the processor and is entrusted not only with the task of generating audio but also with the task of rendering the image on the screen and moving special graphic elements, the "sprites," that represent the player, characters and objects in the scene: this chip is called TIA, an acronym that stands for "Television Interface Adaptor."

**Fig. 1: Atari 5200 SuperSystem (photo: Evan-Amos - source: Wikimedia Commons)**

Leading the development of the TIA is Jay Miner (fig. 2), who would later be known as the "father of the Amiga" for his contribution to the development of the chipset of the eponymous 16-bit platform that would come to life a few years later. Miner's idea is as simple as it is effective: delegate burdensome tasks to specific chips, or co-processors, so as to relieve the CPU's work. The TIA is designed precisely to take audio and video management away from the VCS microprocessor. The console is completed in 1976, although it does not go on sale until 1977. Almost immediately, the development of its successor begins: such haste is dictated by the fact that the commercial life of the VCS is estimated by its own creators at about 3 years (so we arrive at 1979/1980), also in the light of design choices made to contain its production cost, which were reflected in a hardware that was not exactly state-of-the-art and which is expected to encounter rapid obsolescence. In spite of this, the VCS sells well and remains on the crest of the wave well beyond this time frame, certainly not, however, for its technical features but rather for the rich software park it offers its users (you can learn more about this in the article devoted to it in the last issue of our magazine).

Unaware that the VCS will last longer than expected, Atari developers begin work on a project assigned the code "3200." The features must be far superior to those of the VCS so it is decided to use a 10-bit CPU to which several custom chips, each dedicated to a specific task, will be placed side by side. The group headed by Jay Miner focuses on a graphics chip heir to the TIA, called the CTIA, from "Color Television Interface Adaptor," in whose development Doug Neubauer, Steven Mayer and Joe Decuir participate. CTIA, deriving from TIA, has the same limitations as the latter: it was not designed to handle framebuffers, bitmap graphics, and multiple graphics modes. Joe Decuir, Francois Michel and Steve Smith, again under Miner's direction, work on another chip to flank

**Fig. 2 - Jay Miner (source: Alcheron.com)**

the CTIA, the ANTIC, from "Alphanumeric Television Interface Controller," a true programmable graphics processor deputed to the generation of the actual image, capable of autonomously accessing memory ("Direct Memory Access," or DMA) to fetch the data it needs, and which can handle multiple graphics modes, including text modes. Flanking these chips is an integrated chip dedicated to sound generation and control of input systems called POKEY, from "Pot Keyboard Integrated Circuit," to whose development Steven Mayer and Ronald Milner contributed. The new hardware is quite complex and work is proceeding slowly. As if this were not enough, towards the end of 1977 the home computer revolution begins: 3 small computers are put on the market that disrupt the computer market, namely, the Commodore PET 2001, the Apple II, and the TRS-80. Attracted by the possibilities that a computer offers in addition to a simple console, i.e., not only the possibility of playing games but also the possibility of using programs of various kinds, and given the no longer prohibitive purchase price that makes them finally affordable tools for everyone, many buyers turn to these new products. Atari decides to stop the development of the new console and also enter the home computer market: in order not to lose too much ground to the competition and not to throw away all the development work done so far, it decides to recycle what was already ready. There is a problem on the video part: the CTIA is not yet fully developed and some of the graphics modes envisaged by the project are missing, but, in order not to fall too far behind, Atari puts it into production as it is anyway, replacing it shortly afterwards with the full version, called GTIA, from "Graphic Television Interface Adaptor" (fig. 3). The whole thing is assembled around a CPU derived from the MOS 6502 and then put on sale in 1979 as two similar computers but with some differences: we are talking about the Atari 400 and 800.

**The new course**

After the debut of these machines, the console project stands still for some time: sales of the two computers are going very well, and even better is the VCS, which is shattering all records, with sales doubling year after year. Despite these numbers, Atari fears the competition: Mattel, in early 1980, puts out the Intellivision, a console technically superior to the VCS, which is beginning to show all its technological backwardness. Mattel knows that that is the weak side of the VCS and pushes right on the comparison

**Fig. 3 - Atari GTIA (photo: Rodrigo Castro - source: Wikimedia Commons)**

between the rendering of games on its machine and that of the Atari model: on the Intellivision the graphics and sound are close to those of bar games, while on the VCS the graphics are rudimentarily rendered and the sound is limited to simple tones, making it very difficult to convert the latest arcade games. The Pac-Man conversion is representative of these limitations: the hardware differences are there and are all visible in Atari's game, only remotely comparable to the original arcade. Therefore, the project of "Pam," as the console was affectionately called by its developers, a name apparently derived from that of an Atari employee, is taken over. From top management, however, a decision is made that, in the long run, will prove to be a real own goal: although the hardware is identical to that of the 400/800 computers, it is decided to make the two platforms incompatible so that cartridges for home computers cannot be inserted and played on the console and vice versa. The console is equipped with four game ports for as many joysticks and, on top, the port for inserting cartridges. The body of the machine is made of black plastic with a metal band running horizontally across the top and, at the rear, a door that hides from view a compartment where the 2 supplied joysticks can be stored: the very presence of this compartment is the reason why the machine appears generously sized. Internally, the motherboard is based on the same CPU and auxiliary processor triptych found on Atari home computers: the CPU is a 6502 modified to work at 1.79 MHz, internally known as SALLY, while graphics is handled by the 2 GTIA and ANTIC chips, and sound and input by the POKEY. Special joysticks are also developed with a numeric keypad for entering additional commands, starting and resetting games, and, first among all other consoles, even a button for pausing them.

During development, it is initially chosen to market the console under its Pam project name but changing it to PAM, a reverse acronym for "Personal Arcade Machine." However, it is later changed to "Atari Video System X," and several final prototypes of the machine are actually made under this name (some are still in circulation). Eventually, however, it is decided to use the numerical nomenclature adopted for home computers, and to rename the VCS at the same time: the latter becomes the Atari 2600, because during its development that was its internal code, and the Video Sistem X becomes the Atari 5200. This number is not chosen at random: being twice as large as 2600, it is intended to indicate that the console is twice as powerful as its predecessor. It made its debut in November 1982 when the console was introduced as the Atari 5200 SuperSystem and offered for sale at a price

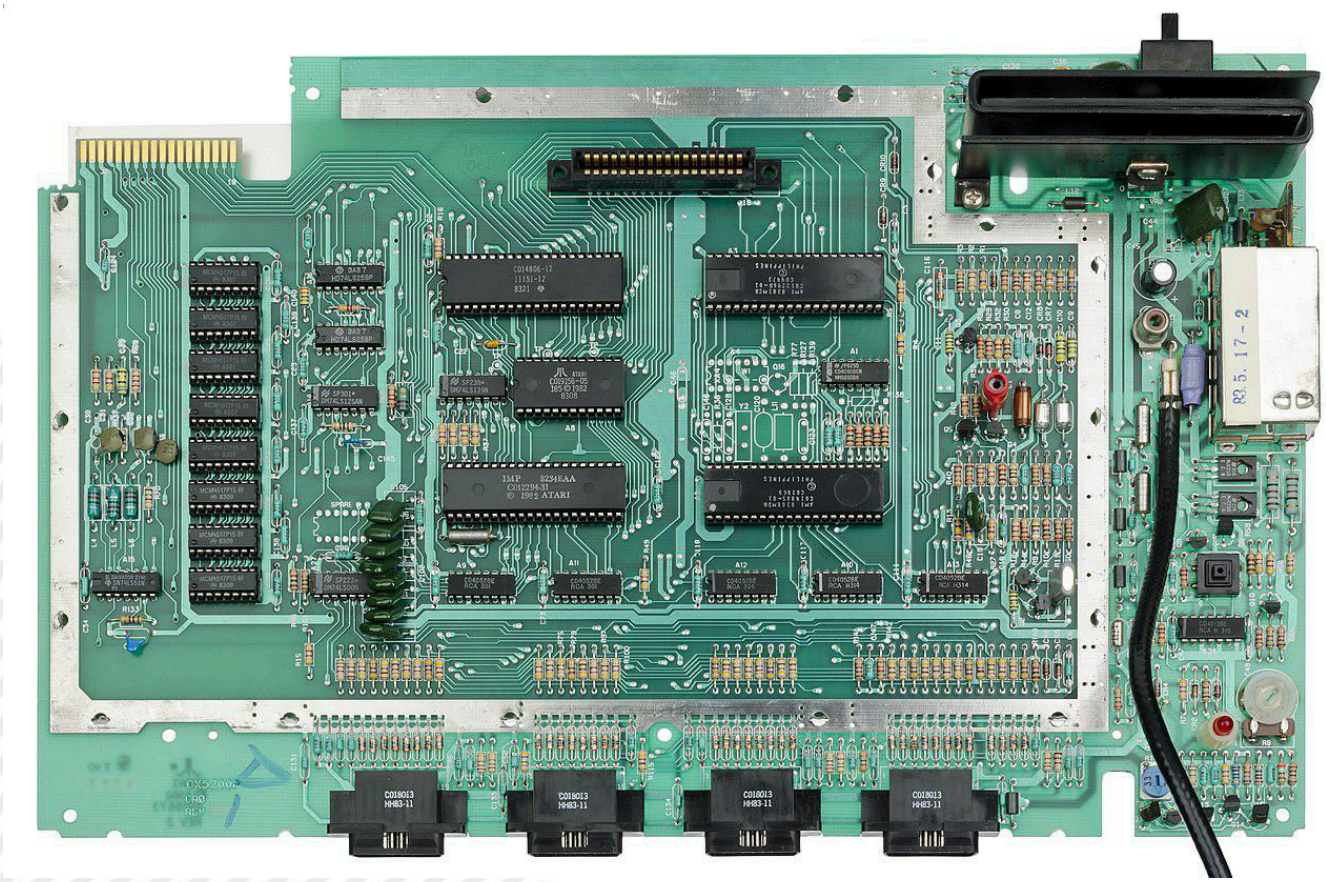


**Fig. 4 - One of the early 1975 prototypes of the future VCS/2600 (source: Wikimedia - author: Chris Platsikoudis)**

of $269.99.

**Specifications**

The console consists of a large plastic body containing the motherboard and, as mentioned, a compartment in which to store the 2 supplied joysticks. Internally we have the same CPU as the home computers, namely the MOS 6502C, where the "C" does not indicate a later revision but only that this is a "custom" chip capable of operating at 1.79 MHz. The video department is handled by the GTIA/ANTIC combo. The GTIA is the full version of the original CTIA design, to which it adds 3 additional graphics modes. The ANTIC is the chip responsible for generating bitmap graphics. It is a true programmable processor by means of mini-programs called "display lists": like a common BASIC interpreter, the processor reads the received display list and generates each line of the screen according to the instructions therein. The ANTIC retrieves the data directly from memory by direct access without CPU intermediation (DMA mode) and then passes the generated image data to the GTIA. The latter applies colors by also managing their hues: because of this mode of operation, there are 16 hues for the 16 base colors, for a total of 256 colors. GTIA also applies sprites to the image: as in TIA, these are called "Player/Missile" sprites and are, respectively, 4 "player" sprites 4 pixels wide each and 4 "missile" sprites 2 pixels wide each. Both have a height equal to that of the video image. There are a total of 14 video modes: there are 6 text modes and 8 graphic modes, the latter with a width of 80, 160 or 320 pixels. The height can be up to 192 pixels although, in overscan, the image can reach a maximum size of 384x240 pixels. The colors that can be simultaneously displayed on the screen are 2, 4 or 16 for 320, 160 or 80-pixel wide images, respectively. Since ANTIC, through display lists, can change register values at each line of the image, there can be up to 256 different colors (16 colors for 16 shades) and 16 different colors for each line. Both horizontal and vertical scrolling is also present.

The POKEY is primarily concerned with sound generation. This chip offers as many as 4 different channels, which can operate either as 4 8-bit or 2 16-bit channels. Each channel allows for independent management of both the volume and the characteristics of the generated wave (square wave with variable "duty cycle" or noise). In addition to this, the POKEY also handles the reading of input devices.

**Fig. 5 - The controller with the control capable of 360-degree movement (photo: Evan-Amos - source: Wikimedia Commons)**

The console has 16 KB of RAM memory, the first two pages of which are used by the console's CPU and BIOS, and the rest available to the game cartridge. Relative to the BIOS, this resides on 2 KB of ROM: the first KB contains the character map (identical to that of the Atari 400/800) while the other is the actual BIOS, referred to here as the "monitor." This consists of the machine initialization routines executed before starting the game on a cartridge, those for handling interrupts, and the "shadowing" of the machine's internal registers. The console can address at most cartridges with 32 KB of ROM: larger quantities require the use of the "bank switching" technique, i.e., a mechanism that selects, among the various banks, the one currently in use.

The joysticks are the hallmark of this machine as well as its Achilles' heel (fig. 5): developed to have 360° movement and not be limited to only the 4/8 directions of the devices in the other systems, they are not, however, equipped with automatic repositioning. This means that, once the pressure to operate them has ceased, the rod does not regain its central position: this creates significant problems in some games, such as Pac-Man, where the control continues to send the character in a given direction unless it is put back into the neutral zone. In addition to this, poor materials are chosen so that cracks in the plastics are frequent, requiring the entire joystick to be rebuilt.

As mentioned earlier, Atari has made changes to the hardware so that games from the 400/800 computers cannot be used on the 5200 and vice versa. This inability is primarily to blame for the BIOS being reduced from 10 to 1 KB, so the functionality built into the computers is not present in the console. The GTIA and POKEY registers are also mapped to different addresses, so converting a game is possible but all values must be changed manually for the software to find these registers at the correct locations. To make matters worse, some of these registers have a different task in the 5200 than in the computer. Finally, the console's particular joysticks are seen as a pair of paddles that provide analog signals requiring a completely different handling of their input than the digital joysticks used on Atari computers.

A special feature of the console is the presence of the Switchbox, an automatic switch for the console signal to be connected behind the home TV (fig 6). Normally other coeval consoles offer a manual switch: it is the user who has to move the switch to select the console or TV cable input. In that of the 5200, a special electronic circuit, as soon as it notices the presence of the video signal on the console input, switches to the latter. To reduce the number of cables, the one connecting TV and console also brings power to the latter.

**Games, accessories and versions**

The console, having had a fairly short commercial life, did not receive much interest from developers. In the 2 years it remained on the market, only 69 games were presented, mostly developed by Atari itself, and mainly conversions of famous bar games such as "Defender," "Frogger," "Galaxian," "Ms. Pac-Man" (fig. 8), "Popeye," "River Raid," "Space Invaders," and "Zaxxon" (fig. 9). Also marketed along with the console is a special controller called "Pro-Line Track Ball" (fig. 7), equipped with a dual alphanumeric keyboard and central track-ball, designed for games such as "Centipede" and "Missile Command." Unlike Coleco, Atari never marketed an accessory to turn the console into a computer, probably due to the fact that this would have eaten into sales of its 400- and 800-series computers. The limited uptake was also reflected in the lack of interest from hardware manufacturers, so that there were few alternatives to its particular joysticks. In 1983, after only one year on the market, Atari introduced an overhaul of the console. Externally, the new model differs in that it has only 2 joystick ports and that the automatic switch has disappeared, replaced by a manual one. This change also involves moving the power supply inside the console itself. Along with the new console Atari also presents an adapter to be able to play 2600 cartridges, something Coleco has done before for its ColecoVision (drawing Atari's ire). In spite of this, sales did not go very well so, in early 1984, Atari announced its discontinuation during the press conference presenting its heir, the 7800 (which, however, due to the division and reorganization of the parent company, would not hit the shelves until 1986).

**Failure**

The reasons for the console's failure are many, and there

**Fig. 6 - The Atari Switchbox, the automatic TV switch (photo: Evan-Amos - source: Wikimedia Commons)**

**Fig. 7 - The Pro-Line Track-Ball (photo: Evan-Amos - source: Wikimedia Commons)**

is no predominant one. Probably the main factor may have been the delay in its commercialization: despite its decent hardware capabilities, the console was meant to debut at least a couple of years earlier, and the hiatus in its development brought it to market already "old." Incompatibility with Atari 400/800 computer games certainly didn't help since absolutely identical platforms were isolated from each other by illogical design choices that ultimately turned out to be real hoes.

Another hard blow came with the release of the ColecoVision in mid-1982, just a few months before the 5200's debut: the latter was supposed to compete with Mattel's Intellivision but ended up with a much more technologically advanced opponent that took away additional potential buyers. Another criticism levelled at it is the questionable choice to include in the sales package the adaptation of "Super Breakout," a game that is now dated and certainly not able to show off the console's capabilities: the comparison with "Donkey Kong" offered together with the ColecoVision is merciless and to the advantage of the latter machine, where the game is reproduced almost identically to the arcade version. In addition, Coleco also understands that the 2600 is not easy to knock off and prefers to offer from the outset an adapter to allow its cartridges to be used with the ColecoVision, thus offering Atari console owners the possibility of not having to dispose of all the games they already own. Atari, for its part, did not understand this point right away and did not offer such an adapter until 1 year after the 5200 was marketed, when the 2600 was registering its highest sales volumes.

Another drawback of the 5200 was its high cost: its initial selling price of $269.99 is certainly not attractive when compared to the ColecoVision's $179. But it also does not hold up against the 2600, which sold for $79 and featured not only better controllers but also a game park consisting of hundreds of titles. Finally, came the 1983 video game crisis to deal the final blow to a console that never shined. It is estimated that in its 2 years on the market, the 5200 sold about 1 million units, the same number as the 2600 did in 1979 alone. Finally, a note on marketing areas. The 5200 was made in NTSC version only, although the ANTIC and GTIA video chips were also produced in PAL versions for Atari computers destined for the international market: because of this, the console had a further limitation on its distribution, restricted to the North American market (U.S. and Canada) only, which adversely affected sales.

Too bad, because the potential was all there, but too high a price tag, some questionable design choices, a delayed release, Atari itself not believing in it too wholeheartedly and continuing to support a console that was now obsolete but a source of sure earnings, and paltry support from software houses reduced it to a mere sidekick, and as such its rapid passage did not leave a significant mark on the video game landscape.

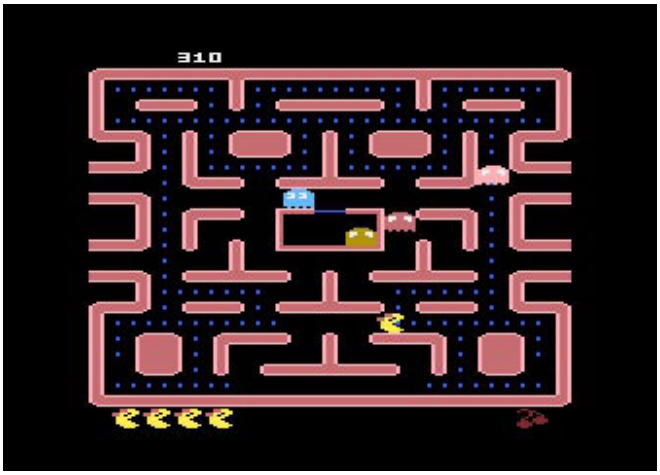



**Fig. 8 - Ms. Pac-Man (source: Mobygames.com)**

**Fig. 9 - Zaxxon (source: Mobygames.com)**

# CALC RESULT - the Spreadsheet for Commodore 64

*by Francesco Fiorentini*

Calc Result is a versatile and powerful spreadsheet produced by Handic Software AB in 1983.
As many as two versions were created at that time to meet the economic needs of users:

| | | | |
|---|---|---|---|
| C,D | **CALC RESULT EXPANDED** (in italiano) Potente programma di pianificazione manage-riale. | C64, SX64 | 195.000 |
| C | **CALC RESULT EASY** (in italiano) | C64, SX64 | 125.000 |

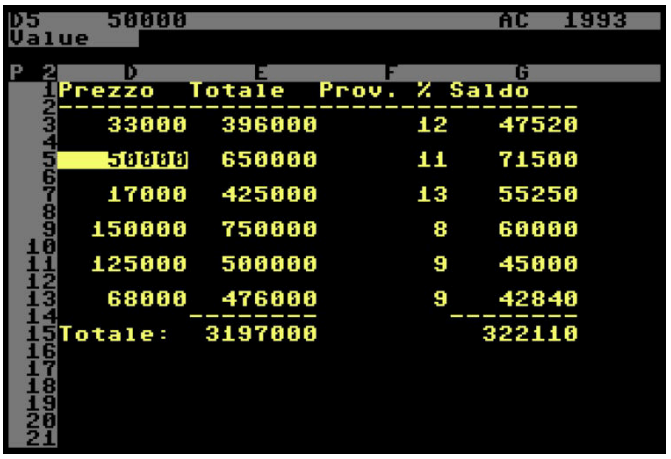
Features of this spreadsheet include 32 pages, horizontal and vertical page splitting, formulas, text, graphs, windows, automatic calculation, error messages, multilingual help screens, storage in DIF file, system commands, disk, edit, print and format.

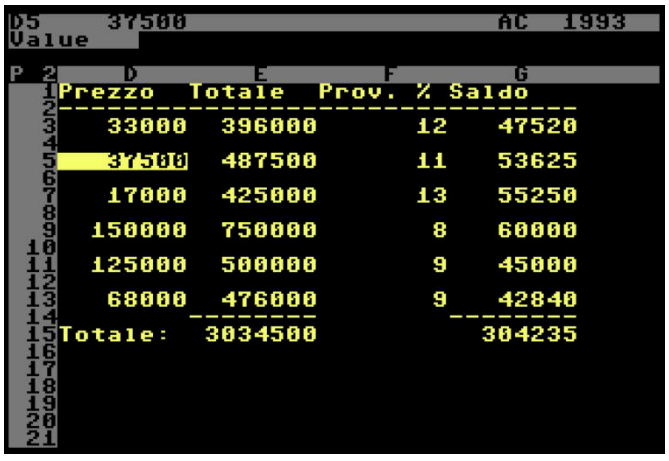
**Handic Software AB**

Handic Software AB, founded in 1952, was originally a radio and telephone manufacturer. Datatronic acquired the company for a Swedish crown in 1980 and used the brand as the official distributor of Commodore computers in Scandinavia. In addition to computers, the company began importing international software titles and in 1983 released the first commercial game developed in Sweden, Space Action. Handic Software AB closed its doors in 1986 after Commodore opened its own office in Sweden.

Calc Result, one of Handic Software's products, is a really well-made and extremely up-to-date spreadsheet with some features that were missing in many other competitors of the time. One for all: with each change of values in the sheet, all formulas are efficiently and accurately recalculated. Oh, and very quickly, too!

See the figures below.




**Special keys of Calc Result**

F7 - Function key F7 is the access key to all Calc Result commands and functions.
SPACE - Enables the difference between a LABEL (text) and a VALUE (a value or formula). To enter text we must first press SPACE and then begin typing the text we want to see appear in the cell.

To copy a formula from one field to another:
- Move to the cell that contains that the formula we want

to copy; in our case E3 containing the formula C3*D3
- press F7
- press E - Edit
- press R - Replicate

function will appear on the screen:

Replicate
SOURCE RANGE TARGET RANGE
FROM ... TO ... FROM ... TO ...

since we started from cell E3, the first part of the formula is already filled in: FROM E3
- press RETURN to confirm TO E3
- Move to cell E13 (see figure below).
- press RETURN to confirm FROM E13
- press RETURN again to confirm TO E13

After that we will have to choose whether to copy the formula in Absolute or Relative mode
- press R for Relative
- press R for Relative

After a quick refresh, the formula present in cell E3 is copied to cell E13 and recalculated with the present reference values C13*D13

Attention: in case we make a mistake in typing a formula or writing a label, we can empty the cell using the function: F7 + B

Exactly like Excel, Calc Result also has a large list of functions that can be used in formulas.
See Figure 2 on page 16.

Let's see some examples together!

**SUM function**

The SUM function, as in Excel, allows you to sum the values specified in the range of cells indicated by the formula. In our case (see figure 1) it will sum the values contained in cells E3 to E13. To enter this formula, simply type SUM(E3:E13) into the target cell. Just like in Excel!

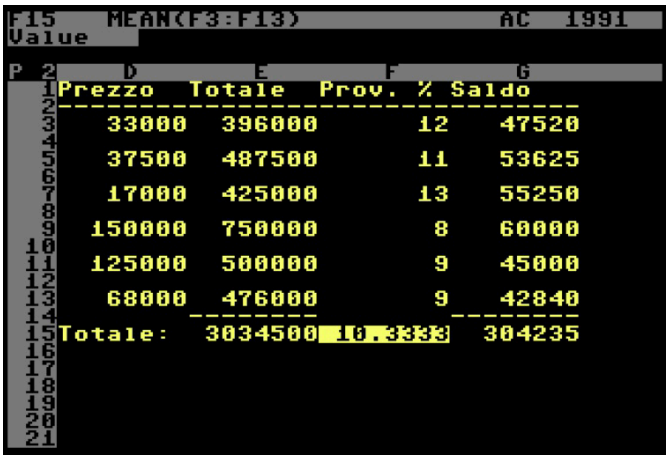
**Fig. 1 -The SUM function**

MEAN function

The MEAN function allows you to find the mean value of a list of values (figure below).

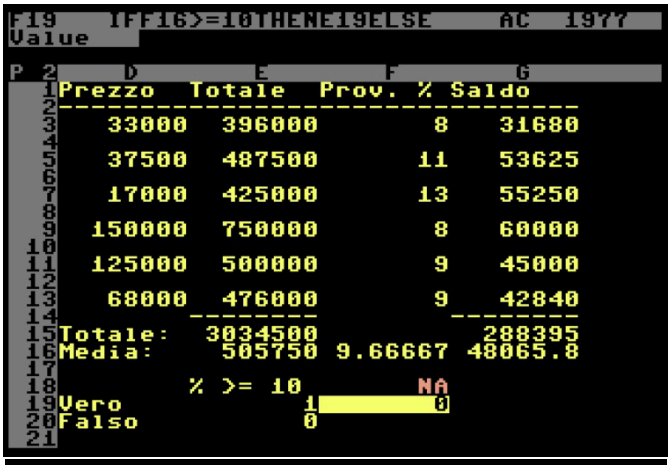
**IF ... THEN ... ELSE**

It is also possible to use the construct IF ... THEN ... ELSE




in our spreadsheets.
Here is an example of this function:
IF F16>=10 THEN E19 ELSE E20

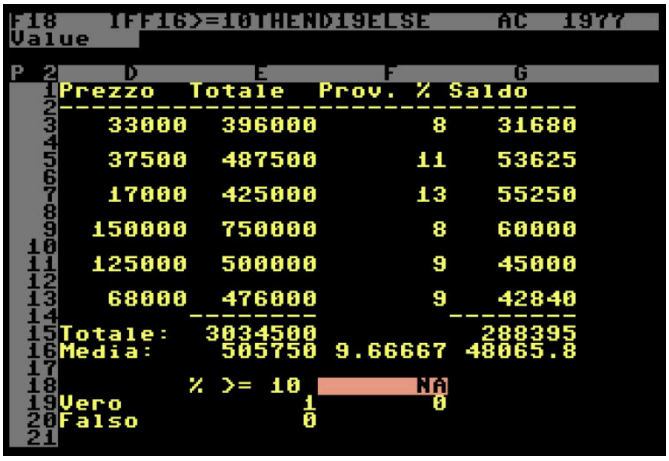
See the figures below for TRUE and FALSE values, respectively.



The same formula can be rewritten:
IF F16>=10 THEN (1) ELSE (0)
Note that formulas can be written omitting spaces:
IFF16>=10THEN(1)ELSE(0)

Spaces that I instead preferred to keep in the article to make it easier to read the formulas.

Unfortunately, it is not possible to use text in the expression (or at least I could not find it), moreover, the same function that points to cells containing text generates an NA error (figure below).



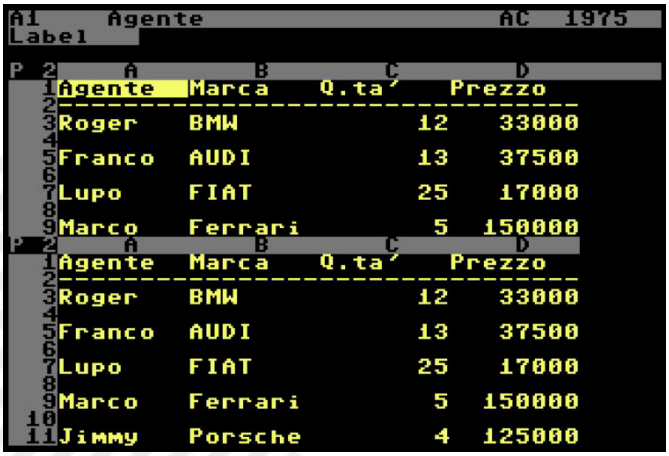

**Screen partitioning**

Calc Result also comes with some features that are definitely amazing for 1983. One of these is definitely the ability to duplicate the screen, both vertically and horizontally.

Move to the beginning of any line and:

- press F7
- press E - Edit
- press S - Split
- press H - Horizontal

This will split your spreadsheet into two independent screens (figure below).




You can move from one screen to another by pressing F2. Note that changing a value in either screen will instantly recalculate the formulas on both. What can I say, definitely amazing.

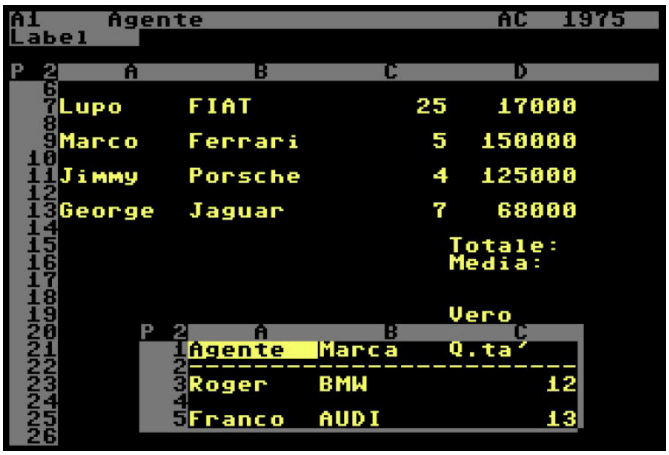
To return to the normal view you will simply press F7 and then L.

**Subwindows**

The surprises don't stop with screen splitting. It is in fact possible to create subwindows in any part of the screen. To try this feature all you have to do is:

- Move to any cell, in our case B21
- press F7
- press E - Edit
- press W - Window
- confirm the starting point of the window: B21
- Type or move to the cell that will border the lower right part of the window: C25

After a short refresh a new window will be inserted at the indicated coordinates (see figure below).




**Charts**

Of course, the ability to generate graphs could not be missing. Unfortunately, I must admit that this feature, although fast and visually appealing, is definitely limited. It is in fact possible to create graphs (histograms) only of the values contained in a given row or column.

In our case we have represented the values in column D.

Move to column D and:

- press F7
- press E - Edit
- press G - Graphic
- press C - Column

Type in the minimum and maximum values of the reference scale: in our case 10000 and 160000

After that, type in the title of your chart: Auto Prices

The result can be seen in Figure 3.

**In conclusion**

I had a lot of fun trying Calc Result because it is a decidedly cutting-edge software for the period in which it was conceived. This spreadsheet proves once again that the Commodore 64 was decisively a computer with a thousand potentials: a gaming tool for millions of kids but also a

**GROUP 1**

COUNT Gives the number of co-ordinates containing a constant or valid formula (without NA or error).

MAX Chooses the largest value within a given area

MIN Chooses the smallest value within a given area

MEAN Gives average value for a certain area

STDDEV Calculates the standard deviation of a given area

SUM Calculates the sum of a given area

*Structure:* FUNCTION (argument, argument, argument.....)

*Example:* MEAN(A1:A5,B1:B5)

**GROUP 2**

NPV To calculate the net present value

*Structure:* FUNCTION (per cent, argument: argument)

*Example:* NPV(.10,C1:G15)

**GROUP 3**

NA For a co-ordinate which is missing a value or a formula which refers to an invalid co-ordinate

PI Gives the constant π value

*Structure:* FUNCTION

*Example:* NA

**GROUP 4**

ABS, ARCCOS, ARCSIN, COS, EXP, FRAC, INT, LN, LOG10, RND, SIN, SQRT and TAN

*Structure:* FUNCTION (expression)

*Example:* SIN(A1+1*10/B7)

**GROUP 5**

IF THEN ELSE

*Structure:* IF expression THEN expression ELSE expression

*Example:* IF A1=10 THEN B1+2 ELSE 5

**GROUP 6**

OR, AND and NOT

*Structure:* FUNCTION within IF THEN ELSE

*Example:* IF A1=10 OR C1=5 THEN B1+2 ELSE 5

**Fig. 2 - The Math functions in Calc Result**

working tool for those who needed to use it in an office environment.

If we then consider that all the things I have described to you work with 64K RAM and a 1MHz processor, well, there is only kudos to be given to the programmers. Definitely amazing!

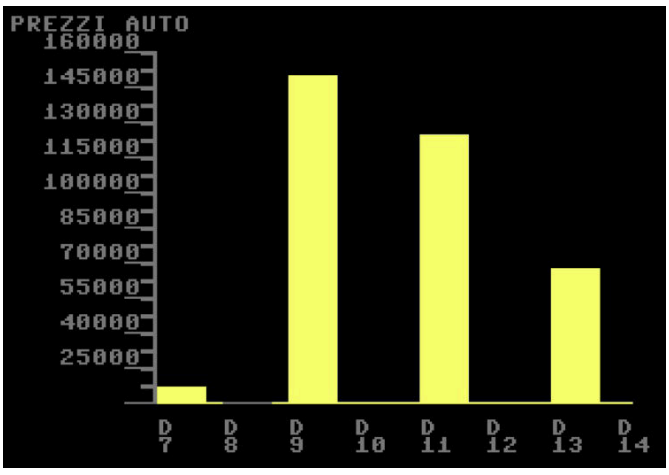



**Fig. 3 - Calc Result's graphs**

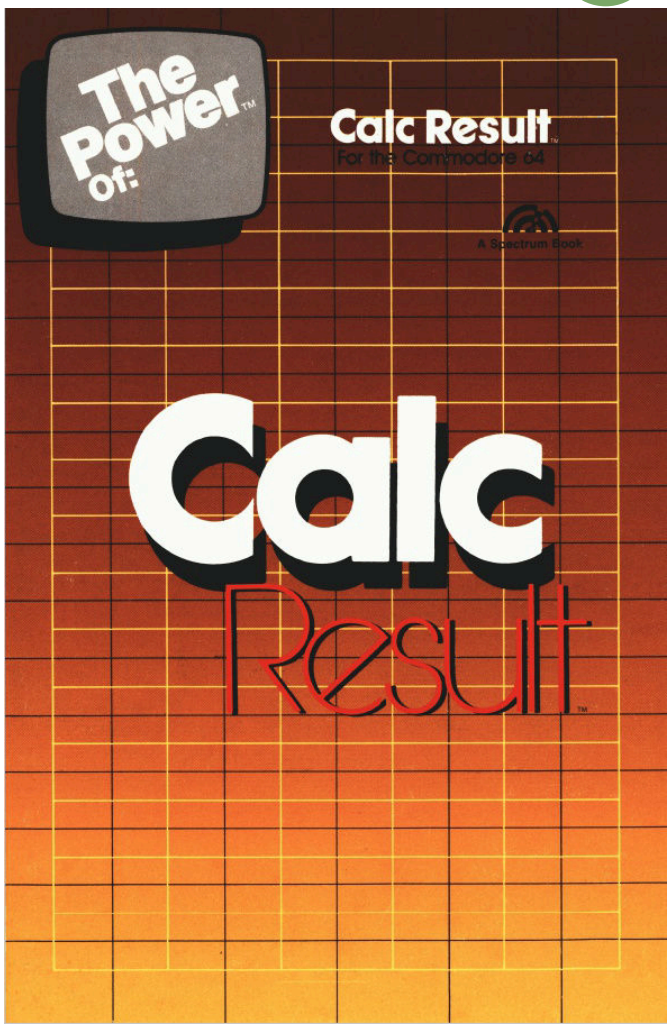



**Fig. 4 - The book: The Power of Calc Result**

**For further study:**

- For those who would like to study **Calc Result** further, I leave the link to its **manual**, which is really well done and detailed:
*https://commodore.software/downloads/download/211-application-manuals/13357-calc-result*

- It is also possible to consult the book **The Power of Calc Result**, which contains a series of exercises useful for understanding how this spreadsheet worked and how it could be used in the Office environment in the 1980s:
*https://commodore.bombjack.org/books/commodore/books/The_Power_of_Calc_Result_for_the_Commodore_64.zip*

- Calc Result, on the other hand, can be downloaded from:
*https://edgeemu.net/details-2855.htm*

# An experiment in Assembly for the C64

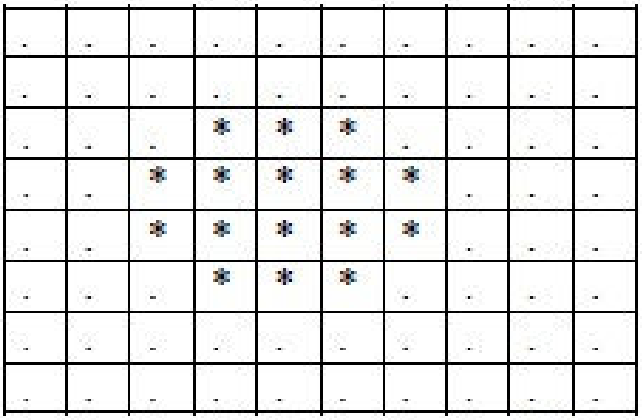
*by Eugenio Rapella*

In my old book, Elementi di base di Probabilità e Statistica, I had presented a program, OPINION, which I had created from a proposal that appeared in the Calculator Recreations column of the June 1985 issue of Le Scienze, the Italian version of Scientific American.
Reviewing that chapter, I came up with this variation where I turned the tables a bit and removed the probabilistic aspect.

At the beginning of the program, the Commodore 64 screen is randomly filled with two symbols "*" and "." representing two states, in some way, contrasting or "in conflict" (could be conservative/reformist; inter and milan supporters; normal skin cells/altered skin cells). The symbol is chosen randomly (with 50% probability "*", 50% probability ".") for each of the 1000 locations on the C64 screen (these are 25 rows by 40 columns). Thus, we begin with a situation in which the two symbols are well mixed, both as numerical consistency and geographical arrangement.

At this point, the main cycle opens: we randomly choose an element of the grid, let's call it alpha (those "on the edge," i.e., those in the first and last rows, first and last columns, are excluded) and count how many asterisks are present among the eight "neighbors," i.e., those in the location immediately north, south, east, west, northeast, northwest, southeast, and southwest. If the number of asterisks is greater than 4, the symbol in alpha becomes an "asterisk" (regardless of its original content), if it is less than 4, the symbol in alpha becomes a "dot" while if it is exactly 4, the alpha content remains unchanged (let's call it the "rule of four"). In other words, the neighborhood situation influences alpha and causes alpha to assume the "location" of the majority of its neighbors (for alphas, edge locations have been excluded so that each alpha has all eight "neighbors"; another way to solve this, and one that I had adopted in the OPINION program, is to "weld" the last row with the first, resulting in a cylinder, and then to connect, by folding the cylinder, the left circle with the right circle. This results in what in geometry is called a "torus"-a doughnut, a lifesaver-and none of the boxes would be on the edge).
Once the transformation is done, the program restarts with the random choice of a new alpha and a new, eventual transformation. What happens in the long run? It is clear that zones where everyone is of the same opinion remain stable because the alphas fished in there maintain their state. In "border" zones, configurations arise that somehow...stabilize.

For example, a configuration such as the one below appears "stable": by counting the neighbors of the various boxes, we see that there will be no change according to the "rule of four."



In the OPINION program I had presented in the book, the game was different: two contiguous symbols with different opinions were drawn and each of the two tried to convince the other to change flags; the neighbors of the pair, ten in number, tried to influence the two, and the probability of success of one or the other was also tied to a probability P, required at the beginning of the program, which provided the "weight" of the neighborhood. In OPINION, we were moving toward totalitarianism in the sense that, in the long run, the grid became totally filled with dots or asterisks (the most interesting aspect of the game was the intermediate configurations and the time required, understood as the number of cycles, to achieve uniformity: the more persuasive the neighbors were, the sooner uniformity of ... opinion was achieved).

In our new version of the game, we let the C64 show us what happens as time goes on.

Here is a possible solution in Basic:

```
5 w=rnd(-ti)
10 poke 53280.0 : rem black border
20 poke 53281.0 : rem black background
30 for t=0 to 999
40 hh=46 : rem dot
50 if rnd(0)<.5 then hh=42 : rem asterisk
60 poke 1024+t,hh
70 poke 55296+t,1: rem white color of characters
80 next t
90 ns=0
100 x=int(rnd(1)*38)+1
110 y=int(rnd(1)*23)+1
120 n=1024+x+40*y
130 if(peek(n-40)=42)then ns=ns+1
140 if(peek(n+40)=42)then ns=ns+1
150 if(peek(n-1)=42)then ns=ns+1
160 if(peek(n+1)=42)then ns=ns+1
170 if(peek(n-41)=42)then ns=ns+1
180 if(peek(n-39)=42)then ns=ns+1
190 if(peek(n+39)=42)then ns=ns+1
200 if(peek(n+41)=42)then ns=ns+1
210 if ns>4 then poke n,42
220 if ns<4 then poke n,46
230 goto 90
```

The FOR-NEXT cycle from 30 to 80 reviews the 1000 screen boxes, the 1024 location corresponds to the upper left corner and is filled with either a "dot" (poke...., 46) or an "asterisk" (poke ...., 42), with probability ½, in instruction 60 when t=0, then it is the turn of 1025, t=1, and so on. At 70 you take advantage of the loop to enter "white" as the color associated with the location (in the C64, the screen character color map begins at location 55296).

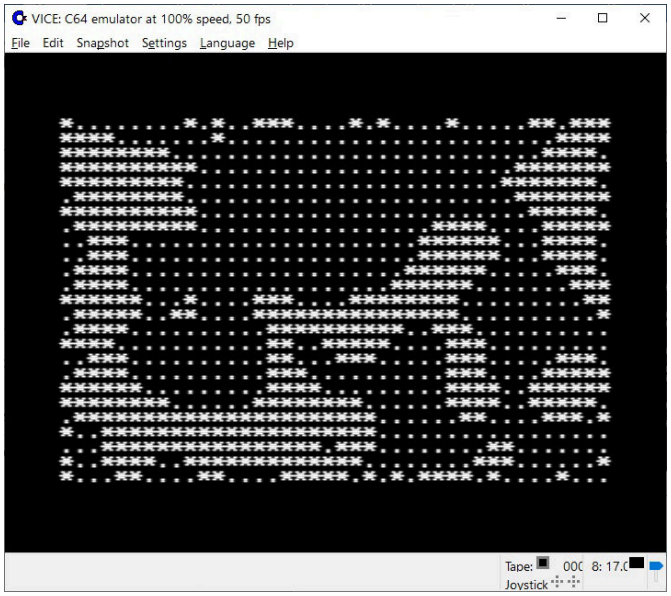
After this preliminary operation, the actual program begins: instructions 90 through 230. ns is the variable that contains the number of neighborhood asterisks; it is initially set equal to zero (instr. 90). The 100 originates a random number that will be the column number of the location chosen for the game, our alpha. There are 40 columns and they are numbered from 0 to 39. Since we want to avoid the first and last, we desire x to be from 1 to 38. The function rnd(1) gives a decimal, random number from 0 to 1: 0 inclusive, 1 excluded: 0≤ rnd(1)<1.

Multiplying by 38 gives a decimal number between 0 and 38 with 38 excluded; taking the integer part int(rnd(1)*38) gives an integer from 0 to 37; with the final "+ 1" we finally arrive at the random value from 1 to 38 (instr. 100). Entirely similar discourse for instruction 110, which gives a random (y) value from 1 to 23 that allows the determination of the alpha row by excluding the first and last (as noted above, there are 25 rows numbered from 0 to 24).
Instruction 120 calculates the number of the C64 screen memory location corresponding to row x and column y, a number that is assigned to variable n.
The eight instructions 130 through 200 all operate in the same way: each checks the contents of the location of one of the 8 "neighbors," if it is an asterisk the variable ns is incremented by one unit so that, when you get to instruction 210, ns contains the total number of asterisks of alpha neighbors, an integer between 0 and 8.

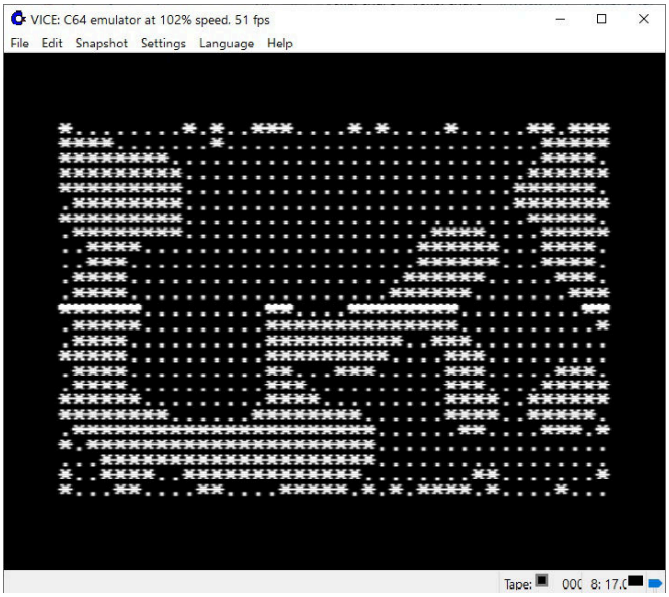
We are at the end: instruction 210, if ns exceeds 4, an asterisk is placed in the location of alpha, n, (regardless of what was there at the time); however, if, instruction 220, the number of neighbor asterisks is less than 4 (which means the dots will be 5 to 8), a dot goes in n (if ns=4, the contents of n remain unchanged). At this point we return to instr. 90 for choosing a new alpha. The program does not end (it will have to be interrupted manually), and on the C64 screen we can watch as the situation unfolds. The symbols that appear on the board will never be changed, but they will affect their neighbors...

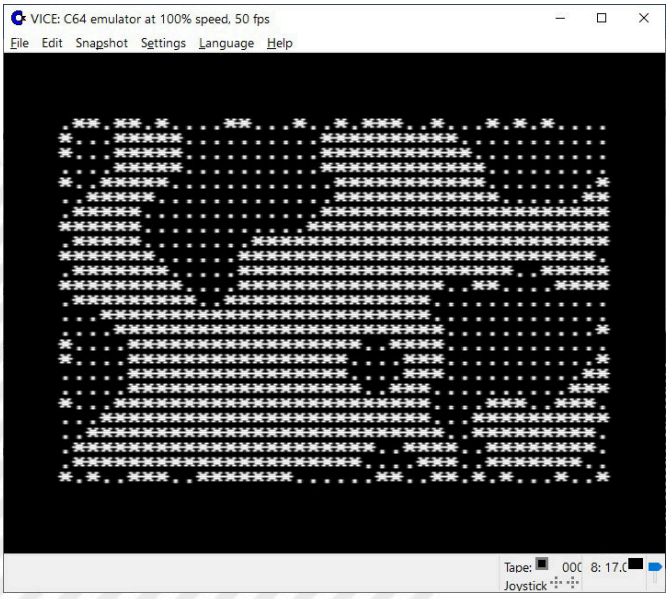
The Basic program is very slow, already creating the initial configuration takes several seconds.
This is what the screen looks like after the start phase:

It takes about 5 minutes to achieve this configuration:

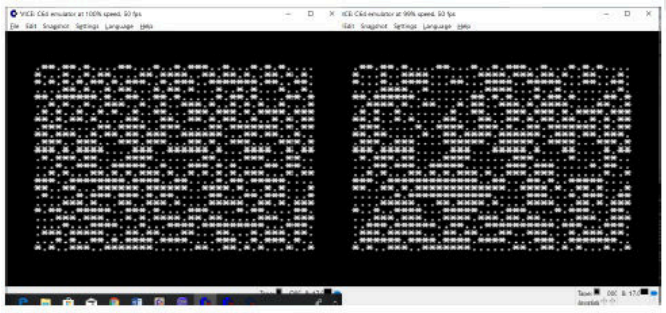


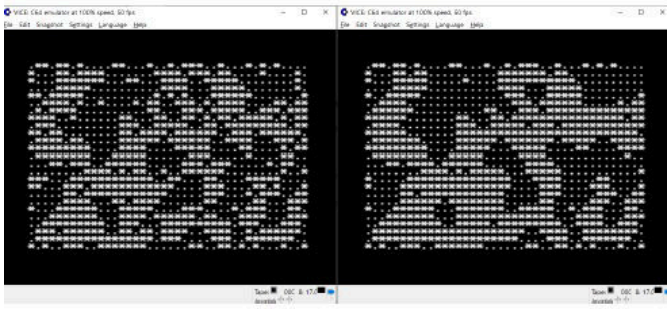

and another 5 or so to get this configuration, which seems to be quite stable:

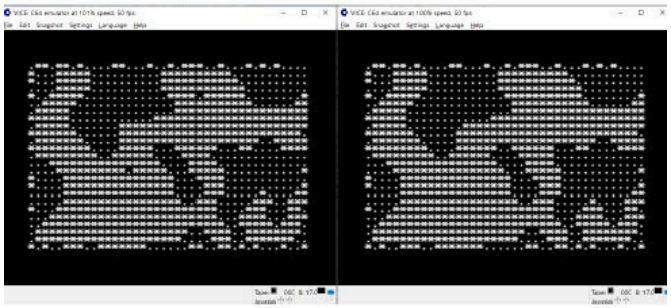



The "final" pattern, which depends on the initial pattern and the symbols along the edge-those that do not change-is not easily predictable; the interesting and fun (for me







it is) part of the program is precisely to watch the pattern evolve to the final pattern.

Why is the processing so slow? Well, our C64's Basic interpreter is certainly not a champion of speed; moreover, if at first we see frequent symbol changes, as time goes on, everything seems to slow down...

The reason is simple: stable zones of increasing area are created, and because the alpha candidate is always chosen at random, it most often happens within these zones.

The Commodore always works at the same speed, but in these areas the symbol in alpha is reconfirmed and, in fact, nothing changes on the screen.

I tried to compile the Basic code (with Blitz Compiler, as I had done, back in the day, with OPINION); of course, the compiled code is quite a bit faster, but... I wanted to do a little experiment with the 6502 Assembly, using CBM Prg Studio (which I also used for the Basic code and, of course, as you can see from the pictures, everything ends up on the legendary VICE emulator).

The code I produced is undoubtedly very, very improvable (it is, indeed, an experiment) but, predictably, the comparison in terms of speed is decidedly in favor of the

latter!

The code I produced is quite rich in comments (alas, in Italian); it helped to clarify some ideas for me, although a few doubts remained.

Here it is:

```
*=$C000; sys 49152
punto=$2e; punto è #46
asterisco=$2A ; asterisco #42
inizioSchermo=$0400 ; 1024 in base 10
inizioColore=$D800; 55296 in base 10
   LDA #0 ; colore NERO
   STA $D020 ; bordo nero
   STA $D021 ; sfondo nero
;Questa Routine (ripresa da Internet) genera un ISSUE casuale da 0 a 255
;nella locazione $D41B. Richiede una inizializzazione ... una tantum:
;***inizializza RND ***
   LDA #$FF ; maximum frequency value
   STA $D40E ; voice 3 frequency low byte
   STA $D40F ; voice 3 frequency high byte
   LDA #$80 ; noise waveform, gate bit off
   STA $D412 ; voice 3 control register
;****************************************
;****** INIZIALIZZAZIONE SCHERMO ******
; Poichè lo schermo contiene 1000 caratteri, viene riempito in quattro
; tornate di 250 caratteri ciascuna. Il contatore è il registro X
; che parte da 250 e si decrementa (dex) fino a zero
; *** i=0 ***
   ldx #250
loop
   lda #asterisco
   sta $0346 ; locazione che contiene il simbolo da stampare
   lda $D41B ; carica in A un RND da 0 a 255
   cmp #$80 ; confronta RND con 128 (a metà tra 0 e 255)
   bcs labb ; se A > 128 va a LABB
   lda #punto ; se A <= 128 "*" viene sostituito da "."
   sta $0346 ; la locazione contiene "." o "*" con prob 1/2
labb
   lda $0346
  sta $0400-1,x ; memorizza il simbolo sullo schermo
   lda #$1 ; colore bianco per i caratteri
   sta inizioColore-1,x
   dex
   bne loop ; se x non è zero torna sopra
   ; *** i=1 ***
   ldx #250
loop1
   lda #asterisco
   sta $0346 ; locazione che contiene il simbolo da stampare
   lda $D41B ; carica in A un RND da 0 a 255
   cmp #$80 ; confronta RND con 128 ecc.
   bcs labb1
   lda #punto
   sta $0346
labb1
   lda $0346
   sta $0400+249,x
   lda #$1
   sta inizioColore+249,x
   dex
   bne loop1
   ; *** i=2 ***
   ldx #250
loop2
   lda #asterisco
   sta $0346 ; locazione che contiene il simbolo da stampare
   lda $D41B ; carica in A un RND da 0 a 255
   cmp #$80 ; confronta RND con 128 ecc
   bcs labb2
   lda #punto
   sta $0346
labb2
   lda $0346
   sta $0400+499,x
   lda #$1
   sta inizioColore+499,x
   dex
   bne loop2
   ; *** i=3 ***
   ldx #250
loop3
   lda #asterisco
   sta $0346 ; locazione che contiene il simbolo da stampare
   lda $D41B ; carica in A un RND da 0 a 255
   cmp #$80 ; confronta RND con 128
   bcs labb3
   lda #punto
   sta $0346
labb3
   lda $0346
   sta $0400+749,x
   lda #$1
```

```
   sta inizioColore+749,x
   dex
   bne loop3
;
; ****** FINE INIZIALIZZAZIONE SCHERMO ******
zpl0=$fb ; locazioni di pagina zero
zpl1=$fc ; per l'indirizzamento indiretto
main
; Creiamo x da 0 a 37
alfa5
    lda $d41b ; RND 0 - 255 ISSUE casuale in A
(Accumulatore)
   and #$3f ; ora RND è tra 0 - 63
   cmp #$26 ; confronta con 38
   bcs alfa5 ; torna ad alfa5 se A>=38
   sta $0342 ; questa locazione contiene x
; Creiamo y da 0 a 22
alfa6
   lda $d41b ; RND 0 - 255
   and #$1f ; RND 0 - 31
   cmp #$17 ; confronta con 23
   bcs alfa6 ; torna ad alfa6 se A>=23
   sta $0343 ; questa locazione contiene y
   sta $0344 ; copia di y
   asl $0344
   asl $0344 ; <- 4y
   lda $0343 ; y -> A
   adc $0344 ; 5y -> A
   sta $0344 ; <- 5y (1 byte, max 115)
   asl $0344 ; <- 10y (1 byte, max 230)
   clc
   lda #$0
   sta $0345
   asl $0344 ; <- 20y
   rol $0345 ; per via del riporto
   clc
   asl $0344 ; <- 40y
   rol $0345 ; per via del riporto
   clc
   lda $0344 ; low di 40y -> A
   adc $0342 ; + x
   sta $0344
   lda $0345
   adc #$0
   sta $0345
   clc
   lda $0344
   adc #$00 ; low byte di inizioSchermo
   sta $0344 ; low di N
   lda $0345
   adc #$04 ; high byte di inizioSchermo
   sta $0345 ; high di N
; ora N contiene 1024+x+40*y, non è il nostro
"alfa", ma
; la locazione una riga in meno e una colonna in meno
; (l'angolo superiore sinistro, rispetto ad alfa).
; In altre parole, anziché scegliere a caso alfa
e contare
; il ISSUE di asterischi "attorno" ad alfa,
conviene scegliere
; a caso l'angolo superiore sinistro del riquadro
(beta) e contare
; gli asterischi presenti in beta, beta+1, beta+2,
beta+40, beta+42
; (beta+41 corrisponde ad alfa) e infine beta+80,
beta+81 e beta+82;
; in questo modo si eseguono solo addizioni.
; Copiamo N (beta) nella locazione utilizzata
successivamente:
   lda $0344
   sta $033e
   lda $0345
   sta $033f
; Conteggio del ISSUE di asterischi: valore
aggiornato nel registro X
   ldx #$0
; $033e (low byte) e $033f (high byte) contengono
la locazione dello schermo
; oggetto del controllo (contiene un asterisco?).
All'inizio è "beta", poi
; viene aggiornata sommando 1 (per ottenere
beta+1), poi sommando un altro 1
; (per ottenere beta+2), poi si addiziona 38 (per
ottenere beta+40), poi si
; addiziona 2 (per ottenere beta+42), poi gli
aggiornamenti proseguono con +38,
; +1 e infine +1. La parte di codice tra "+++++"
e "-----" viene dunque
; ripetuta 8 volte: all'inizio l'INCREMENTO viene
caricato in A e la subroutine
; "gamma" si incarica di aggiornare $033e e $033f.
; +++++ (inizio: siamo a NW di alfa, angolo
superiore sinistro rispetto alfa)
; N,S,E,W = nord, sud, est, ovest, NE = nord-est ecc.
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$00
   lda (zpl0),y
   cmp #asterisco
   bne alfa0
   inx
alfa0
; -----
    lda #$01 ; INCREMENTO +1 (siamo a N di alfa)
   jsr gamma
   ; +++++
   lda $033e
```

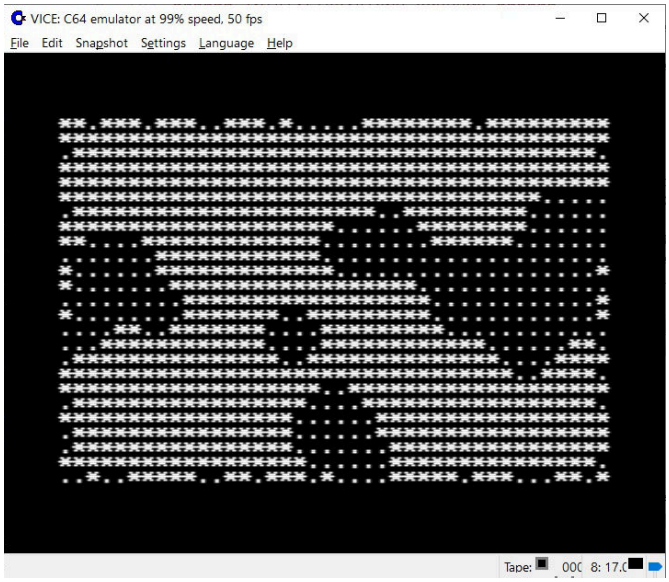
```
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$00
   lda (zpl0),y
   cmp #asterisco
   bne alfa1
   inx
alfa1
; ------
   lda #$01 ; INCREMENTO +1 (siamo a NE di alfa)
   jsr gamma
   ; +++++
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$00
   lda (zpl0),y
   cmp #asterisco
   bne alfa2
   inx
alfa2
; -----
   lda #$26 ; INCREMENTO +38 (siamo a W di alfa)
   jsr gamma
; +++++
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$00
   lda (zpl0),y
   cmp #asterisco
   bne alfa40
   inx
alfa40
; -----
   lda #$02 ; INCREMENTO +2 (siamo a E di alfa)
   jsr gamma
; +++++
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$00
   lda (zpl0),y
   cmp #asterisco
   bne alfa42
   inx
alfa42
; -----
   lda #$26 ; INCREMENTO +38 (siamo a SW di alfa)
   jsr gamma
; +++++
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$00
   lda (zpl0),y
   cmp #asterisco
   bne alfa80
   inx
alfa80
; -----
   lda #$1 ; INCREMENTO +1 (siamo a S di alfa)
   jsr gamma
; +++++
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$00
   lda (zpl0),y
   cmp #asterisco
   bne alfa81
   inx
alfa81
; -----
   lda #$1 ; INCREMENTO +1 (siamo a SE di alfa)
   jsr gamma
; +++++
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$00
   lda (zpl0),y
   cmp #asterisco
   bne alfa82
   inx
alfa82
; -----
; Ora il registro X contiene un intero da 0 a 8
pari al numero
; di asterischi che circondano la locazione "alfa".
; se X<4 in "alfa" va un puntino; se X>4 va un
asterisco.
; La locazione "beta" era conservata in $0344
(low) e $0345 (high),
; copiamo i valori in $033e e $033f in modo di
```

```
sfruttare la subroutine
; "gamma" per addizionare 41 e ottenere la locazione "alfa"
   lda $0344
   sta $033e
   lda $0345
   sta $033f
   lda #$29 ; corrisponde a 41 in base 10
   jsr gamma
   cpx #$4
   bcs alfabis ; salta se X >= 4
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$0
    lda #punto ; (dunque, se X<4 in "alfa" va un puntino)
   sta (zpl0),y
   jmp out ; torniamo all'inizio
alfabis
   cpx #$4
   beq out ; salta se X=4
; se siamo qui significa che X>4 e in "alfa" va un asterisco:
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$0
   lda #asterisco
   sta (zpl0),y
   out
   jmp main
   rts
; subroutine gamma: somma il contenuto di A al contenuto
; di $033e (low byte) e di $033f (high byte)
   gamma
   clc
   adc $033e
   sta $033e
   bcc teta
   lda #$01
   clc
   adc $033f
   sta $033f
   teta
   rts
; fine subroutine gamma
```

The game/program is open to some variations. One possibility is to favor one of the two symbols in the initial setup. If, for example, we make it so that, at the beginning, the probability of "*" is 75% versus a 25% for "."

```
loop
   lda #asterisco
    sta $0346 ; locazione che contiene il simbolo da stampare
   lda $D41B ; carica in A un RND da 0 a 255
  cmp #$40 ; confronta RND con 64 (a 1/4 tra 0 e 255)
   bcs labb ; se A > 64 va a LABB
  lda #punto ; se A <= 64 "*" viene sostituito da "."
  sta $0346 ; la locazione contiene "." con prob 1/4
labb
```

the final configuration will also contain many more asterisks than dots:




It is also possible to change the "rule of four" to favor one of the two symbols.

Another possibility is to keep the "rule of four," but to make it work ... in reverse: if in the neighborhood the number X of asterisks is greater than four, the box in the center becomes a dot; if X < 4 the box in the center becomes an asterisk. The more neighbors of one type, the more the subject in the center switches to the other side (a behavior analogous to the ANTIVOTERS program I had considered in the book).

How does the initial configuration evolve with this “contrary-minded” rule? There is only to try, although we can imagine that concentrations of the same symbol will be avoided (a "checkerboard" arrangement of the two symbols would be stable).

The changes to be made in the Assembly code are minimal,

just swap "dot" and "asterisk" in the code after the counting of the number of asterisks (in the program in Basic, just swap 42 and 46 in instructions 210 and 220 with each other):

```
cpx #$4
   bcs alfabis ; salta se X >= 4
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$0
   lda #asterisco
   sta (zpl0),y
   jmp out
alfabis
   cpx #$4
   beq out
   lda $033e
   sta zpl0
   lda $033f
   sta zpl1
   ldy #$0
   lda #punto
   sta (zpl0),y
out
```

Without varying the names of the constants, different ASCII codes can be associated with "dot" and "asterisk" in order to achieve more ... blatant graphical effects. At the beginning of the Assembly code we modify the two lines:
dot=$2e; dot is #46
asterisk=$2A ; asterisk #42
at
dot=$20; now the code is 32: space, square "empty""
asterisk=$A0 ; here the ASCII code is 160: square "full"
(in the Basic code we would replace 46 with 32 and 42 with 160 in instructions 40, 50 and 130 through 220) and run the program in Assembly with the variation... "contrary-minded".

Here are some results with the new and old symbols:

# Graphics... What a passion!

## Part IV - The KLA graphics format and how to deal with it

*by Gianluca Girelli*

As we know, appetite comes when eating, and so here we are again on these pages talking about graphics for 8-bit systems, primarily C64 and C128. Having learned in Issue 40 how the arrangements that help form a multicolor bitmap image are structured on some of the older Commodore machines, let us now see how to import that data into a high-level program to re-edit the image; it will then be easier and more effective to prepare better arrangements for our future projects.

As mentioned earlier, there are nowadays extremely powerful tools for the creation and manipulation of graphic images which, mimicking to perfection the mechanics of bitmap handling according to Commodore specifications, allow one to create redefined character sets, sprites and figures on one's PC in a simple, intuitive and modern way. One such tool is Multipaint (Figs. 1 and 2), and through it, images can be created, exported, and re-imported for most retrocomputers by emulating the color limitations we have discussed in the recent past (e.g., the maximum number of colors within an 8x8 pixel matrix). At the present time, Multipaint supports the following formats: Commodore 64 high resolution, Commodore 64 multicolor (Fig. 3), Commodore Plus/4 Hires, Commodore Plus/4 multicolor, ZX Spectrum, ZX ULAplus, MSX 1 (Fig. 4), Amstrad CPC mode 0/1 (and 0 with Overscan), Sinclair QL and Atari ST (Fig. 5) low resolution. (Figs. 3 - 5) The C128 is not directly supported, but on the aforementioned Issue 40 we learned that the files that make up the bitmaps are absolutely compatible between C64 and C128 as long as the loading address of "screen.dat" is changed.

After solving the problem of how the files that make up our multicolor image were handled and in what part of the system's memory they are stored, I then wondered how I could import these files within Multipaint. The main reason is due to the fact that there is no way to totally compensate for the difference in the number of colors of any image taken from the Internet (or from your camera) and the limited palette (only 16 colors) of our reference systems. For this reason, in the article on No. 14, the good Marco urged to choose images with a modest number of colors; otherwise, the result of the "transformation" process will result in a product with decidedly grainy graphics, as we shall see later.

Studying Multipaint's documentation, I ascertained that in "C64 multicolor" mode, the program exports and imports files in ".kla" format; this format is the one used by a very famous program of the 1980s, called "Koala Painter," which allowed drawing via a graphic travolette and associated optical pen (Figs. 6 and 7). Developed in 1983 by Koala Technologies Corporation and initially intended for school use, it was produced in various versions for the Apple II, the TRS-80 Color Computer, the 8-bit Ataris, the Commodore 64 (of course) and also for IBM PCs.

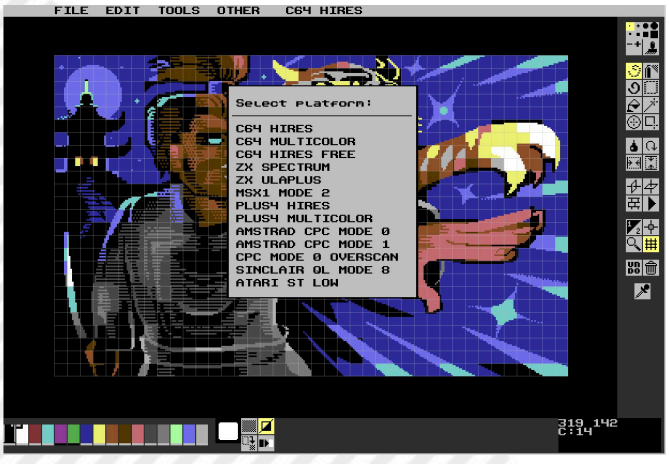



**Fig. 1 - Multipaint main screen**

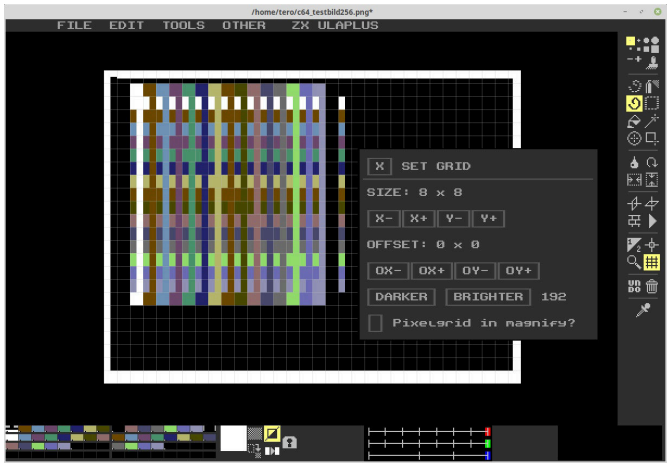



**Fig. 2 - Multipaint options screen**

**Fig. 3 - C64 screen by Davide Bucci**

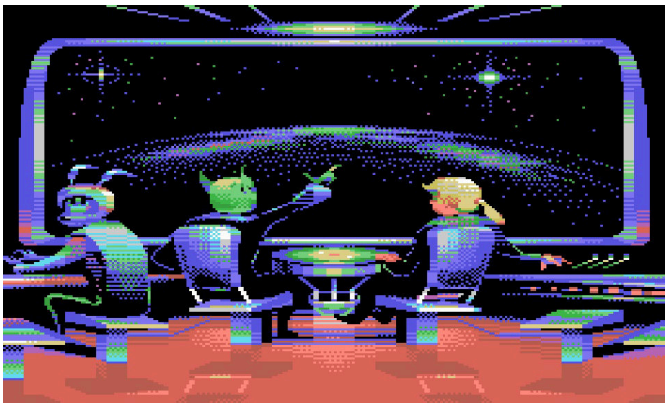


**Fig. 4 - MSX screen by DrTerrorz**

In particular, the Commodore 64 version of KoalaPainter used a fairly simple file format that corresponds directly to the way bitmap graphics are handled on the computer: a two-byte loading address, followed immediately by 8,000 bytes of bitmap data, 1,000 bytes of "Video Matrix" data, 1,000 bytes of "Color RAM" data, and a byte for the background color. Since this is but a simple binary file of 10,003 bytes, I therefore thought of writing a program that would allow one to extract the data of interest from a single ".kla" file or, conversely, to reassemble one from the ".dat" files generated by Marco's program. For the purpose, I used the "Hollywood" language, an M.A.L. (Multimedia Application Layer) based on the LUA language and initially developed on the Amiga which I have discussed in previous articles.

In the first case (data extraction) the procedure is as follows:
- open the .kla file for reading
- copy the file inside a binary string
- open hires.dat in write
- write in the file the address $2000 (standard address)
- write in hires.dat the 8,000 bytes of data structure starting with the third one (the first two are the load address of the .kla), then close the file
- open screen.dat in write
- Write into the file the address $1c00 (standard address for the C128) or $0400 (for the C64)
- write in screen.dat the next 1,000 bytes and close the file
- open colors.dat in write
- write in the file the address $d800 (standard address)
- write in colors.dat the next 1,000 bytes and close the file
- finally close .kla

Note that there is actually one last byte missing from the roll call, corresponding to the one containing the background color, which is the same for all cells. You can set it later as you wish with a POKE or COLOR command (for C128 only). Also note the use of strings to work on binary files: since in Hollywood there is no limit to the maximum number of characters in a string, this approach allows a more flexible approach to copy/paste groups of data. Finally, note that all addresses should be entered in little-endian format (Note [1]).

Reassembling a ".kla" file from the set of ".dat" files is even easier and is sufficient:

Village
Turns: 1
The village is characterised by its picturesque centre, where beautiful half-timbered houses are illuminated by the moon and the leaded windows of the inn to the north reflect the glow of a cosy fire. To the east there is a stable and to the west you can see a church, southwards you may return to the forest.
You notice: a well and a locked stable door.
>_

**Fig. 5 - Atari ST screen by Stefan Vogt - Rabenstein**

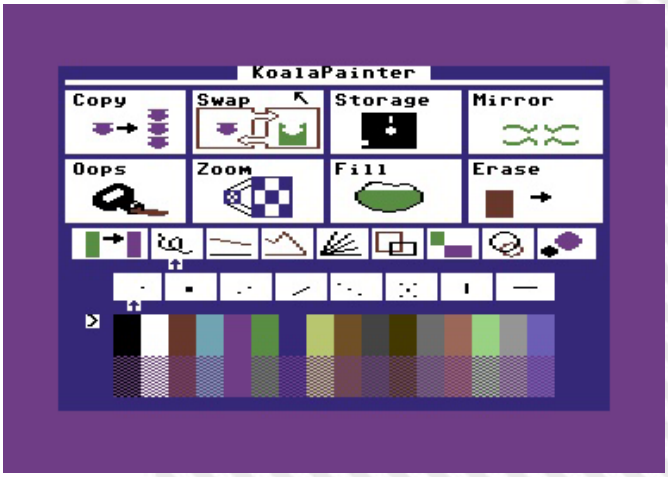



**Fig. 6 - C64 Koala Painter**

**Fig. 7 - KoalaPad**

- create/open write the .kla file
- Write there the address $6000 (standard address)
- Copy to it, in sequence, hires.dat, screen.dat and color.dat omitting for each file the first two bytes
- add a byte with value $00 (black background)
- close all files.

I reproduce below the source code of the two programs for possible in-depth study pointing out that since Hollywood is a proprietary language, it is necessary to purchase the framework in order to use and modify them. A precompiled version for free use will be made available via the Retromagazine World site as soon as possible.

```
;kla_extraction code
;by Gianluca Girelli 2022

;get kla file size
size = FileSize("test1.kla")
OpenFile(1, "test1.kla")

;copy binary data into alphanumeric string
data$ = ReadBytes(1, size)
;extract raw binary data from string to form
dedicates files
hires$=MidStr(data$,2,8000,  #ENCODING_RAW)
screen$=MidStr(data$,8002,1000,
#ENCODING_RAW)
colors$=MidStr(data$,9002,1000,
#ENCODING_RAW)

;open first file in writing mode, add loading
address, copy raw data, close file
;addresses are added in little-endian mode
OpenFile(2, "hires.dat," #MODE_WRITE)
WriteByte(2,ToNumber("00",16))
WriteByte(2,ToNumber("20",16))
WriteBytes(2, hires$, 8000)
CloseFile(2)

;open second file in writing mode, add
loading address, copy raw data, close file
OpenFile(3, "screen.dat," #MODE_WRITE)
WriteByte(3,ToNumber("00",16))
WriteByte(3,ToNumber("1c",16))
WriteBytes(3, screen$, 1000)
CloseFile(3)

;open third file in writing mode, add loading
address, copy raw data, close file
OpenFile(4, "colors.dat," #MODE_WRITE)
WriteByte(4,ToNumber("00",16))
WriteByte(4,ToNumber("d8",16))
WriteBytes(4, colors$, 1000)
CloseFile(4)
CloseFile(1)

;kla_inception code
;by Gianluca Girelli 2022

;get single file size (actually useless,
since they are already known - 80002, 1002,
1002)
size_hir = FileSize("hires.dat")
size_scr = FileSize("screen.dat")
size_col = FileSize("colors.dat")

;open/create .kla file ...
;then add loading address in the first 2 bytes
OpenFile(1, "test2.kla," #MODE_WRITE)
WriteByte(1,ToNumber("00",16))
WriteByte(1,ToNumber("60",16))
```

Fig. 8 - Babil junior

Fig. 9 - Babil junior on Commodore 64

```
;open hires.dat and add it to .kla, after
removing the initial address
OpenFile(2, "hires.dat")
hires$=MidStr(ReadBytes(2, size_hir),2,8000,
#ENCODING_RAW)
WriteBytes(1, hires$, 8000)
CloseFile(2)

;open screen.dat and add it to .kla, after
removing the initial address
OpenFile(3, "screen.dat")
screen$=MidStr(ReadBytes(3, size_scr),2,1000,
#ENCODING_RAW)
WriteBytes(1, screen$, 1000)
CloseFile(3)

;open colors.dat and add it to .kla, after
removing the initial address
OpenFile(4, "colors.dat")
colors$=MidStr(ReadBytes(4, size_col),2,1000,
#ENCODING_RAW)
WriteBytes(1, colors$, 1000)
CloseFile(4)

;add last byte to .kla (backgroung color)
- total 10003
WriteByte(1,ToNumber("00",16))
CloseFile(1)
```

At this point we have made all the tools necessary for our purpose, and so it is time to test them. Specifically, I chose to work on an image taken from an old anime from the 1970s: Babil Ni Sei (Babil 2nd, known in Italy as Babil Junior).

The process was as follows:

- I downloaded an image from the Internet, converted it to 320x200 bitmap and processed it with Marco's program (see RMW #14). In Fig. 8 you can see the original image and, in Fig. 9, the one processed and loaded onto the C64;
- using the .dat files, I obtained, through my code, a .kla file that I loaded into Multipaint (Fig. 10);
- later I corrected and reworked the image with Multipaint, a process not yet completed (Fig. 11). The image data can later be extracted again for inclusion in a complete project for C64/128.

We have once again reached the conclusion of our journey and, once again, we have a new tool at our disposal for our adventures in the "retro-verse." As always, the solution presented does not claim to be the best but I hope it has tickled your fancy to experiment and keep your passion alive. See you in the next issue!

**RESOURCES**

http://multipaint.kameli.net/
https://en.wikipedia.org/wiki/KoalaPad#KoalaPainter
https://www.hollywood-mal.com/
https://github.com/marcus73/retromagazine_03

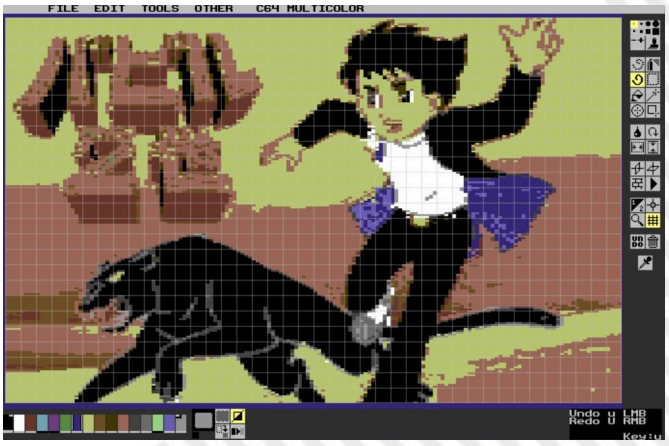



Fig. 10 - Original Babil junior

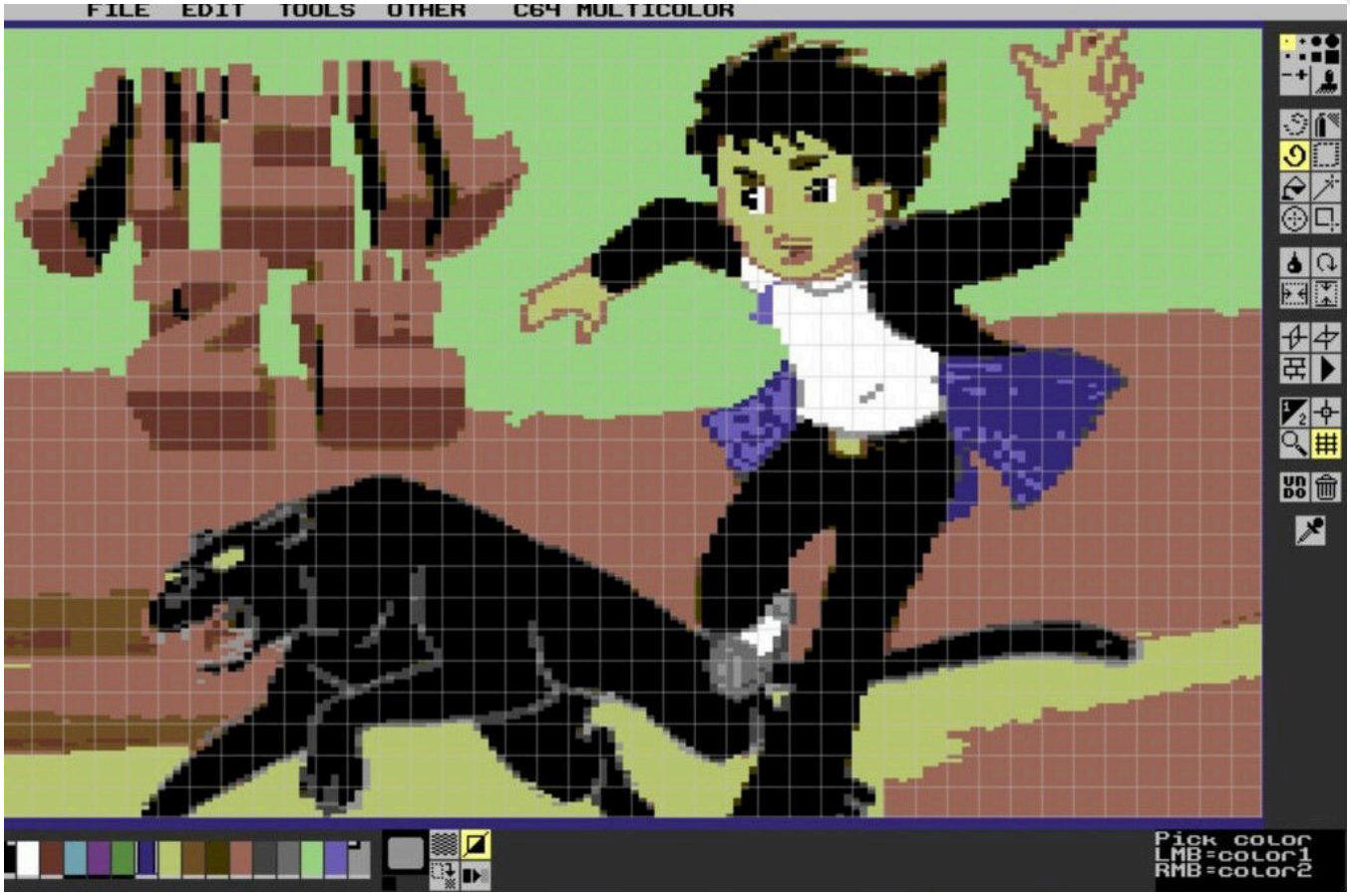



**Fig. 11 - Edited Babil junior**

https://github.com/marcus73/retromagazine_04
http://commodore128.mirkosoft.sk/vic-iie.html
https://it.wikipedia.org/wiki/Babil_Junior

**BIBLIOGRAPHY**

Graphics... What a passion! RMW#13 Marco Pistorio
Graphics ... What a Passion! Part 2 RMW#14 Marco Pistorio
Introduction to graphics on Commodore C128 RMW#21 Gianluca Girelli
Graphics ... What a Passion! Part 3 - RMW#40 by Gianluca Girelli, with contributions from Marco Pistorio
C64 Programmer's Reference Guide - Copyright (C) 1982 by Commodore Business Machines, Inc.
Compute's_Mapping the 64 and 64C - Sheldon Leemon, Compute! Publications Inc 1984,1987
C128-128D System Manual(IT) - Commodore Italiana SpA, 1985
C128 Service Manual - CBM, Inc 1987
C128 Beyond the Handbook - Rita Bonelli, Jackson Editions 1986
Mapping the C128 - Ottis R. Cowper, Compute! Publications Inc. 1986

**NOTES**

[1] The processor is "little-endian" so the byte that is stored first in the locations designated as address is the least significant byte (least significant) and the byte that comes next is the most significant byte (most significant). Although visually to us humans it appears that the bytes are in reverse order, it actually makes sense that the least significant byte is at a lesser address and the most significant byte is at a greater address. In "big-endian" processors, the opposite happens. Source: Simone Bevilacqua, "The Basics of Programming.

**THANKS**

The author would like to give special thanks:
Tero Heikkinen (Dr. TerrorZ), for Multipaint and MSX screenshot;
Davide Bucci, for the C64 screenshot, taken from "The Queen's Footsteps."
Stefan Vogt, for the ATARI screenshot taken from "The Curse of Rabenstein."

# Optimizing MSX BASIC and Assembly support

*by Germán Gómez Herrera - Systems Engineer - Spain*

In the last article devoted to this topic, we talked about how to improve execution speed performance when writing a program in Basic language.

This time we will try to improve the performance of a Basic program by using Assembly, sometimes incorrectly also called machine code, where the Basic program can call any routine in Assembly.

Let's take the game example from the last article again. The full game is playable, but if you remember, the background was a static stage.

The question is, wouldn't a dynamic background be better? And if so, could we make a dynamic background? For example, a scrolling graphical background. The answer is yes, and the way to do it is given below.

A first approach, would be to write a rotational scroll. This is easy in the MSX system (1) because the routine only has to handle the name table (see the listing below) and also takes up very little memory space (2).

```
tableequ    01800h + 32
bufferequ   0c100h
v2requ      059h
r2vequ      05ch

scroll:

 ld hl,buffer
 ld b,23

loop:

 push bc

 ld a,(hl)
 ld d,h
 ld e,l
 inc hl
 ld bc,31
 ldir

 ld (de),a

 pop bc

 djnz loop

 ld hl,buffer
 ld de,table
 ld bc,768-32

 call r2v

 ret

setTable:

 ld hl,table
 ld de,buffer
 ld bc,768-32
 call v2r

 ret
```

But this method shows a simple and unrealistic visual effect. In fact, when you observe moving objects from a distance, you can see that they move with different speeds. This is called the parallax effect (3).

Parallax effect

We already know what we need to do, but how do we code it? This routine is more complex than the previous one.

On the one hand, the routine must handle each object,

**Fig. 1 - 8 stars background**

in this case the stars, one by one. On the other, we have to use something like indexes to point to each object.
Writing the program in assembly can be quite difficult, so it is more convenient to use a compiler.
If a compiler is used, programming performance will be much better.
In addition, the compiler can be used optimally.
For example, you can use tables instead of conditional constructs, do not use struct-type variables but individual vectors, and so on. These tips will improve the speed of execution of the routine in machine code.

The compiled language of choice is C (4) because it is closer to the computer. For example, pointers can be used to point to the memory address.

In this language, we have written three functions: the first to set objects, another to move them, and the last is an auxiliary function.
In addition, we examined the output assembly code to correct it.

Based on the last advice, it is much better to use tables instead of conditional constructs; below I show an example where each star is painted with a color based on its distance in depth.
To move stars, you can assign a shift in this way:

```
color=tcol[i];
x[i]-=(8*(color==15)+4*(color==5)+2*(color==4));
```

Where color is the color code of the star.
But it is more appropriate to use tables in this other way:

```
int tcol[3] = { 15, 5, 4 };
int tspeed[3] = { 8 ,4 ,2 };
x[i]-=speed[i]);
```

In the latter way, the size of the assembled code is smaller and thus the execution speed performance will be higher.

The complete code is given in the listing below:

```
int nstar=16;

int tcol[3] = { 15, 5, 4 };
int tspeed[3] = { 8 ,4 ,2 };
int i, ctmp;
int *px, *py;

int x[32],y[32],col[32],speed[32];

initStars(){
   for(i=0;i<nstar;i++){
      x[i]=rnd(256);
      newStar();
   }
}

moveStars(){
```

**Fig. 2 - 16 stars background**

```
    for(i=0;i<nstar;i++){
       pset(*(px=&x[i]),*(py=&y[i]),1);
       if ((*px=*px-speed[i])<0){
          *px=255;
          newStar();
       }
       pset(*px,*py,col[i]);
    }
}

newStar(){
    y[i]=rnd(180)+10;
    col[i]=tcol[ctmp=rnd(3)];
    speed[i]=tspeed[ctmp];
}
```

Once the code is obtained in Assembly, all that remains is to encode it into the data in Basic and call the routine from the Basic program via the right address.

In addition, the number of stars to be handled by the routine must be considered. The number is sixteen by default, but you can change it up to 32 stars by managing the ST variable. Also, it should be noted that if there are many stars, the execution speed of the routine will be lower.

The differences can be seen in Figures 1 to 3:
- Figure 1. Background with 8 stars
- Figure 2. Backdrop with 16 stars
- Figure 3. Backdrop with 32 stars

**Fig. 3 - 32 stars background**

**Fig. 4 - The game with parallax scrolling**

Overall, the result is as follows: A full game with parallax scrolling and up to four sprites simultaneously in play as shown in Figure 4.

The self-executable disk image with the game is attached as usual:
https://retromagazine.net/download/msx_demo_2.dsk

As is often said:
Practice makes perfect.

Have fun!

### References mentioned in the article

1. https://en.m.wikipedia.org/wiki/MSX
2. https://map.grauw.nl/articles/scrolling.php
3. https://en.m.wikipedia.org/wiki/Parallax
4. https://en.m.wikipedia.org/wiki/C_(programming_language)

# A few words with the genius of Kobe

*by Takahiro Yoshioka – translation Carlo Nithaiah Del Mar Pirazzini*

Microsoft founder Bill Gates once said, "Of all the people I have ever met, Kay (Kazuhiko) is probably the most like me." Kazuhiko Nishi, born in Kobe, Japan, in 1956, is one of the pioneers of the era that led to the birth of personal computers.

He once headed Microsoft in Japan and is the inventor of the MSX standard, one of the first and most beloved computers of the 1980s.

At 66, Nishi-San has no plans to give up his commitment to innovation. He is active in many projects, from building cheaper supercomputers to founding a university in Japan. The Open University of Catalonia recently invited Nishi to Barcelona as part of the program's 25th anniversary celebration in Computer Science, Multimedia and Telecommunications.

There, he presented the updated model of the MSX microcomputer, more than 30 years after it became available to the general public.

Today Nishi is a star, an inspiration to all programming and computer engineering enthusiasts. A successful entrepreneur who finds his true calling in playing with lines of code, he has the prophetic aura of someone who understood, before anyone else, that technology must get as close to people as possible.

In the interview, we learned some additional information from Mister Nishi that had not been revealed until now. We hope that this article will serve as a reference point for those who are still hesitant to support this crowdfunding.

**Taka: - Mr. Nishi is kind of a legend to me. A creator and innovator who worked in the PC/IT industry for 30 years Why did he stop developing MSX 30 years ago?**

Nishi: -In 1983, IBM was promoting its own computer, but it was still very expensive and I felt the need to create a cheaper domestic solution. That's how MSX was born. However, I was in Microsoft and one company could not

**Fig. 1 - MR. Nishi and the prototypo of the MSX0**

support two competing products. Therefore, the IBM machine became popular because it was more powerful, so investment increased and prices decreased. By 1993, MSX was no longer profitable, so I stopped developing it.

**T: - Why didn't your product become the international standard to which you aspired?**

N: - The United States was the market for Commodore computers. Great Britain was [ZX] Spectrum's. I was very lucky in Spain, Holland, Italy and Latin America. I was in charge of design, and designers generally don't worry about marketing. Now I am in charge of everything for the next MSX model and will travel all over Europe to talk about the project.

**T: - Does the success of retrogaming and the new nostalgia market have anything to do with its return?**

N: - No, there are new things here. In 1980, the personal computer was born. In 2000, the SMARTPHONE. Every 20 years a new type of computer is born. From 2020, it's time for the IoT (Internet of Things). My idea is to produce a very compact MSX that is connectable to IoT sensors- this is the new MSX0, priced at about $150. Next year I will release the MSX3, which will be connectable to the TV set. And then, the MSX Turbo, which will be a super-

computer. I want to offer a simpler solution than our competitors. I use the [programming] language BASIC, which is old, but still good and easy to use. Also, I don't like video games.

**T: - After so many years, his passion does not seem to wane. What is the secret?**

N: - Everyone is born with a God-given mission ... but no one knows what it is. Of course, it would be easier to be born with a piece of paper that tells you what the mission will be! Instead we are all on a journey to find it. I don't know if it's my mission, but maybe my talent is engineering. At Microsoft, I did two great things: MS-DOS (Microsoft Disk Operating System)-with GW-BASIC and BASIC extensions-and defining the Windows keyboard and mouse, as well as connecting the CD-ROM. This is all thanks to me.

After Microsoft, I made a lot of money with my company working on CPUs and also founded the worldwide community for the MPEG video compression format. When I turned 60, I wondered what I was going to do next. I realized that my next challenge would be IoT.

**T: - Did you ever regret leaving Microsoft in 1985?**

N:- In life there are positive or negative changes, but they are always your decision. I was very close to Bill Gates- I controlled Japan on behalf of the company and only reported to him. But there were other bureaucrats who wanted to take my place. Often, when companies get big, they get filled with ambitious bureaucrats. I decided not to fight and left. If I had stayed in Microsoft, I would now have at least one-tenth of what Bill Gates owns. I suffered for about 10 years after that decision. But then I recovered and took my company public. I made about $300 million: isn't that enough? For a while I thought it wasn't enough, but it was. I mean, I have four helicopters, a Rolls-Royce, a Bentley. So, I'm happy.

**T: - Is this your idea of happiness?**

N: - Everyone is looking for happiness. And then people feel that money can buy happiness. There are so many ways to be happy that I've discovered: achieving your dream, being famous, doing a good job, becoming famous. Tons of money. But if you only seek money, power and fame, you have no limits. With possessions, you have to be mentally aware, you have to have gratitude. Monks in Tibet are happier than people who live with a lot. Perhaps the coexistence of these things can make people extremely happy.

**T: - Back to the present: what do you think about the Big Tech crisis?**

**Fig. 2 - Nishi, Bill Gates and Paul Allen**

Fig. 3 - MR. Nishi

Fig. 4 - The MSX0

N: - Elon Musk laid off three-quarters of Twitter's employees, but the company is still running. Everyone knows that Big Tech is full of lazy workers because the companies are so profitable that they can afford it. Management has been very generous over the years: whether you work or not, it doesn't matter, relax. Now, because of Elon Musk's extreme discipline and attitude, these companies are firing people as a gesture, to show that they are making an effort. The truth is that in every company, school or community, at least 10 percent of the staff is unproductive.

**T: - What do you think about ChatGPT?**

N: - My professor at MIT, Marvin Minsky (one of the pioneers in AI research), always said it's better not to do business based on AI. But now it is becoming very popular. The most advanced country is China, while the United States is desperately trying to catch up. I don't know what will happen, but ChatGPT is still very, very stupid.

When I read the texts he produces, I always laugh. That said, it's only a matter of time -- machines will become very smart in 10 or 20 years. Personally, though, I am not very interested. I still have about 20 years of work left, and I want to devote half of it to IoT and half to super-computers.

**T: - What do you have in mind for super-computers?**

N: - Many people read and hear about super-computers, but almost no one has touched them, because they are incredibly expensive. If you could produce one for the price of a car, everyone could use it. We might not have one in every home, but we would have one in every university or laboratory. Incidentally, super-computers still use the Fortran 77 language, which is 45 years old. We need to bring them closer to people so that they can think of new languages for AI.

**T: - Is he still learning?**

N: - I had to drop out of university because I had to start my own company. When I could, I tried to return, but the university system did not allow it. Eventually, I finished my Ph.D. at the age of 60 and was offered a teaching position at the University of Tokyo. I have been teaching for five years. Now I have to retire, but I have decided to build my university by investing all the money I have. I hope to achieve this goal by 2025-that will be my biggest challenge. Innovation does not depend on physical age, but on mental age and heart. Youth is not a specific period of life: you can stay young by changing your way of thinking.

# ListAmiga - The best 10 platform games on Amiga

*by Beppe Rinella*

Dear brothers and sisters who, like me, drink soda found around in the hope that it will give you incredible powers, welcome back to the usual ListAmiga!

Our dear old Amiga gave us hundreds of wonderful games from very different genres. Some types of games found particular fortune on Mother Commodore's machine, among them are undoubtedly platform games.
Amiga offered all of us a great number of platformers of truly sublime quality, so much so that compiling this list was really difficult for me, the illustrious exclusions are really many, maybe too many!

The concept of a platform game is really very simple and basic and can be summarized roughly as follows: we impersonate a character who has to get to the end of the level by jumping on platforms, avoiding enemies and dangers of various kinds.
The beauty of such simplicity is the possibility of interpreting the genre in very different ways. It can be developed by focusing more on adventure, exploration, or shooting. The mood can range from total cheerfulness, with levels full of colors and funny characters, to a more serious atmosphere.

In the list I offer, I have tried to include not only the best titles in terms of quality, but trying to include titles that show the different declinations possible for this genre.

We therefore start with

**THE BEST 10 PLATFORM GAMES ON AMIGA**

**FIRE AND ICE**

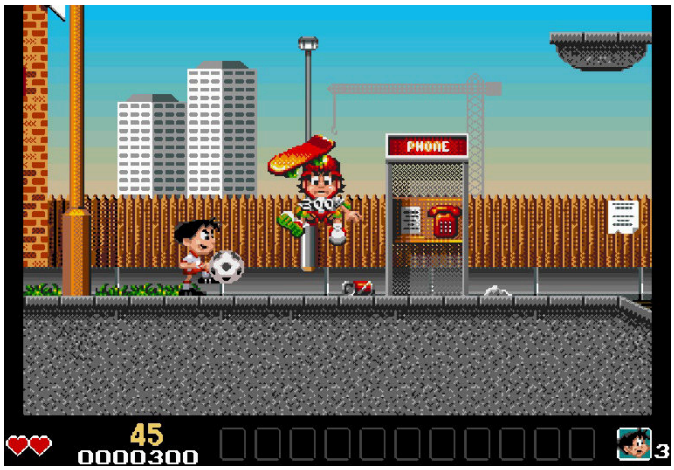
Let's start with a bang.
In Fire and Ice we will be called upon to impersonate Cool Coyote, a blue Coyote who will have to face 7 levels with very different settings. To complete each level, it will be necessary to find the different pieces of the key needed to be able to open the exit. Each piece is guarded by one of the enemies in the level, but we will not know which ones, which is why we will have to eliminate them all. To

**Fire and Ice**

do this, our only weapon are ice balls with which we can momentarily hibernate the unfortunate person on duty, and then we can conveniently smash him into a thousand pieces!
We will start our adventure in the "Arctic" level, then move on to the countryside, the interior of a castle, the underwater level, the rain forest, an Aztec temple, and finally ancient Egypt where we will have to face the bad guy Suten.
The levels are definitely extensive and all to explore, chock-full of enemies and assorted dangers.
Graphically Fire and Ice is a real delight, starting with Cool Coyote animated perfectly, then moving on to the cute and sympathetic enemies only in appearance. The environments then are perfectly characterized and varied.
At the bottom of the screen then is a small map showing us where we are in our adventure, with a reflection effect on the water that is a real nice touch.
Fire and Ice is a joy to play, be prepared, however, for a difficulty level that is not exactly benign, nothing excessive but quite challenging.
I've always loved Fire and Ice and I still do, it's a really exquisite game and it just spews love from every pore!

**SOCCER KID**

We talked about it on these pages in the last issue, in the list of the best soccer games, despite the fact that it is a platformer, and what a platformer.
As our little champion, we will have to retrieve the pieces of the World Cup (shattered during the attempted theft




**Soccer KID**

by trophy-collecting aliens) scattered in five different countries-England, Italy, Russia, Japan and the USA.

To accomplish our mission we will have as a weapon our trusty ball with which to beat up enemies who try to get in our way. The protagonist is a real phenomenon with the ball between his feet, so we will be able to perform dribbles, backhands, headers, and more in addition to being able to use our ball as a trampoline to perform great leaps.

Each country we will be visiting consists of three levels that end with the ever-present final boss. Scattered throughout the different levels we will also be able to collect figurines, eleven for each country; completing the collection will give us access to bonus levels.

Soccer Kid really has it all, graphically it doesn't miss a beat, our hero is richly animated, environments and enemies are excellently crafted and characterized, the whole thing flows without the slightest uncertainty. As long and challenging as it needs to be, and damn entertaining.

In short, there is really no reason not to play and replay Soccer Kid!

**SUPERFROG**

You can't talk about platformers on the Amiga without mentioning Superfrog, and in fact here it is.

The story tells of the most classic of witches turning the most classic of princes into a frog, while also kidnapping the equally classic princess. Our (not yet) hero, sad and disconsolate, swallows a drink (Lucozade, the game's sponsor) thanks to which he becomes Superfrog, complete with a fluttering red cape and off to rescue our beloved.

Nothing particularly original, but there is no need to be original when, gripping the joystick, one is faced with a title of such fine workmanship.

**Superfrog**

Considered (and presented, at least by the trade journals at the time) the answer to Mario and Sonic on the Amiga, it is neither, it is Superfrog and it is a wonder like that.

Beautiful to look at, thanks to graphics that are a delight and flawless, the music by a really on form Allister Brimble is a pleasure to listen to.

The gameplay is over-the-top and the challenge level thankfully lower than the Team17 standard. The levels we will face (6 worlds for a total of 24 levels) are extensive and full of secrets to unearth, the bonus level with the slot machine is a really nice interlude between levels, the shooter level "Project F" a real nice touch.

Superfrog is an ode to the simplicity and purity of the platforming genre; it doesn't want to be anything more, doesn't try to revolutionize anything. Everything it does, however, it does damn well.

Impossible to play Superfrog without a big smile on your face.

**TURRICAN 2**

An absolute masterpiece, I am absolutely certain that if

**Turrican 2**

you are reading these lines you have played it and lots of it too.

The plot, as is often the case, is quite simple and classic: the galactic emperor "The Machine" destroys the spaceship Avalon 1 on which our hero Bren McGuire, the only surviving survivor who, wearing Turrican armor sets out in search of revenge.

Turrican 2 is a decidedly different platformer from those we have seen so far (and will see later), here we shoot and take no prisoners, thanks in part to the offensive potential we are equipped with. Along our path we will find and upgrade devastating weapons, holding down the button on our trusty joystick will start a series of shots in rapid succession that form a very powerful beam that we can direct 360 degrees, but without being able to move. We can then transform into a devastating indestructible spinning ball that sows highly destructive mines. The only flaw in this configuration is the poor maneuverability (intentional of course, otherwise it would really be too easy!). The ability to fire a laser that exterminates all enemies on the screen, as well as the "berserk" mode, i.e., "ball" configuration and total devastation through the simultaneous use of any weapon possible!

In the subject of "level design," Turrican 2 would easily be a subject for examination. The 12 levels, divided into 5 different settings are huge, not only action then but a lot of exploration that yes requires some dedication, but gives satisfaction as few games can do, I speak in the present tense not by accident.

Graphically, Turrican 2 borders on perfection and does not show the slightest uncertainty, even in stages where the number of enemies on the screen is truly remarkable. Variety is then also ensured thanks to the beautifully crafted shot'em up levels, with a little homage to the classic R-Type as we set off in our spaceship. These are things that warm the heart.

The sound deserves a separate chapter, the name Chris Huelsbeck is a guarantee of sublime quality, the music that accompanies us throughout our adventure is simply wonderful, listen to the OST to believe.

Turrican 2 succeeds in the not-so-easy task of improving on its predecessor in every respect, no easy feat given the very high quality of the first chapter.

We are in the presence of a genuine masterpiece, I have said it before but I will say it again, a piece of video game history, not only on the Amiga.

**James Pond 2**

**JAMES POND 2 - CODENAME ROBOCOD**

I'll drop the mask and say it right now, this is one of my video games of the heart, so many are the wonderful memories it brings up in me. But that is not why it is on this list, because James Pond 2 is really a great platformer.

Starting with the story, again nothing exceptional: Dr. Maybe, villain of the first chapter, takes over Santa's toy factory, imprisoning the poor penguins. Our task will be to free them by going around the length and breadth of the 50 levels the title consists of.

James Pond 2 is a simple and essential platformer; all we will be able to do to counter the many enemies present will be to avoid them or jump over them in full Mario style. Our armor (hence the reference to the title's "Robocop," whose rearranged soundtrack we find in the presentation) also allows us to stretch disproportionately upward until we find a platform to cling to, allowing us to reach unspeakable heights.

Although the gameplay may seem all too mundane (find the penguin - exit the level), variety is provided by the size of the levels, which are large and full of secrets to discover, along with power ups of various kinds such as wings for...well, flying, an umbrella for gliding as well as a number of vehicles we can use here and there. Cars, airplanes and a formidable flying tank!

Graphically James Pond 2 is a real treat, while not highly spectacular everything is adorable, including the enemies that are not actually adorable. The same can be said of the music, which is excellent and perfectly in theme.

The levels are varied and include different settings: candy, teddy bears, music and more.

The controls are simple and precise, the difficulty level is not too high, but completing the fish-robot adventure is far from easy, considering also the length and the lack

**Leander**

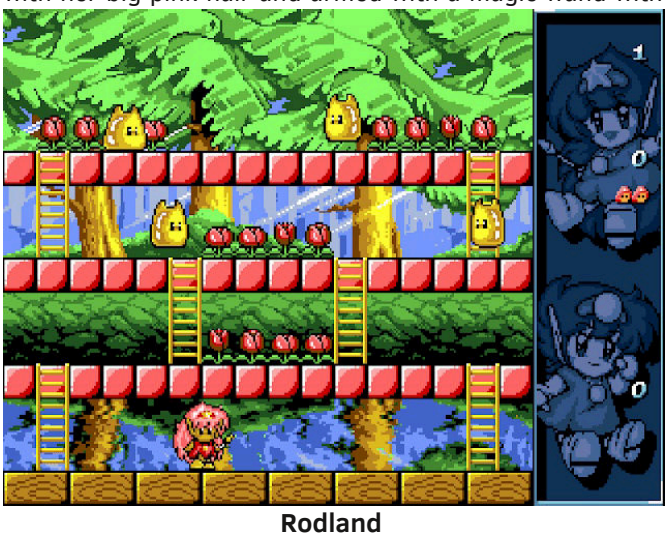
of saves or passwords that would allow us not to have to start all over again every time we insert the floppy (so much beauty condensed into just over 1Mb!).
James Pond is pure joy, simple, straightforward, challenging, colorful, and great fun-what more could you ask for from a platformer worthy of the name?

**LEANDER**

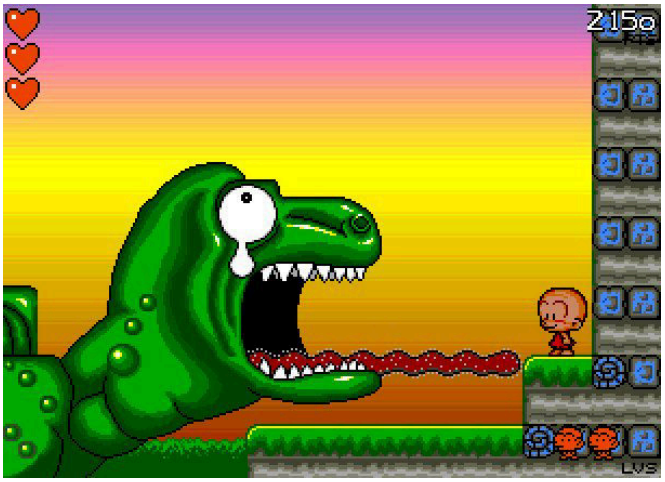
And here we come to Traveller's Tales' resounding debut feature, Leander, a product of an over-the-top quality.
We get it by now, when we talk about platformers the plot is certainly not what matters most, very briefly: as knight/prince Leander we will have to face and defeat the evil Thanatos to free our beloved princess.
Leander combines the platforms proper to the genre with a massive dose of adventure and exploration. We will thus find ourselves wandering the length and breadth of the three worlds divided into 21 total levels, making our way through fierce and highly spectacular enemies in search of a specific item that will allow us to move on to the next level.
We would thus be immersed in the wonderful fantasy settings of this masterpiece, full of atmosphere and just as many dangers, accompanied by our trusty sword with which we will face all sorts of fantastic creatures. Around the levels we would find valuable coins expendable in the stores we would encounter, where we could buy increasingly powerful armor and weapons.
Technically Leander is a gem in every respect, graphically it does not have the slightest smear, we will often face monsters as colossal in size as they are lethal, the highly evocative settings do the rest, creating a truly immersive atmosphere.
On the audio front, the only flaw is that we will be forced to choose between music or sound effects, the latter of which do their job admirably, while the music is the work of the magnificent trio of Matthew Simmonds, Timothy Wright and Jon Burton, authors of the grand symphonies that accompany our hero.
Experiencing Leander's adventure is a real pleasure, the difficulty is excellently calibrated, the rhythms never frenetic as much as reasoned but not without epic confrontations.
Leander is a title to experience at least once in a lifetime, but once you finish it I bet you won't be able to resist the urge to return inside its wonderful world.

**RODLAND**

There is a genre of platformers whose undisputed progenitor is Bubble Bobble, whose characteristics basically are single-screen levels where you eliminate the enemies that populate it, so as to move on to the next level, clean up again, and so on.
Rodland is practically a newer Bubble Bobble than its historic spiritual father, delightful and quite challenging. Again, let's hurry up with the plot question right away, again nothing striking: Tam and Rit, the protagonists, are two fairies who live peacefully in their beautiful enchanted village, until their poor mother is kidnapped by the bad guys led by a giant minotaur, of course our heroines will immediately leave to rescue their imprisoned mother on top of a tower.
Rodland is the conversion of Coin-Op that appeared in 1990 in the arcade and then was converted from 1991 on various platforms. On the Amiga the conversion is really flawless, which is certainly not a frequent occurrence for conversions from Coin-Op for the Commodore machine. Compared to Coin-Op the number of levels is less, some minor graphical rework is there but nothing striking.
We will therefore find ourselves impersonating Fairy Tam, with her big pink hair and armed with a magic wand with



**Rodland**



**BC Kid**

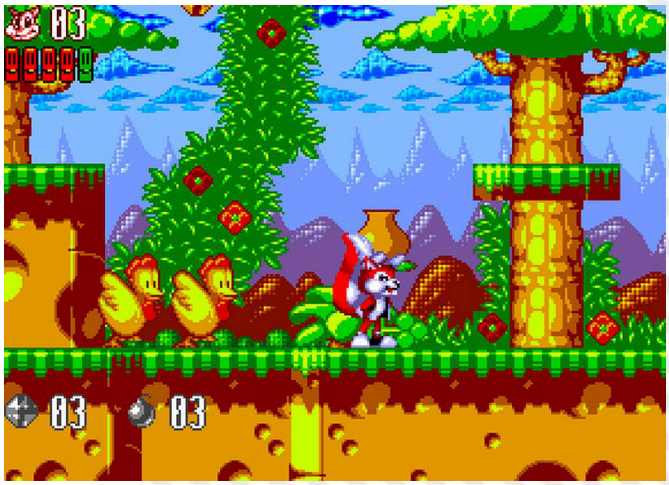
which to eliminate the enemies that haunt the levels. To do this we can either hit them by pushing them back or imprison them in a small magical cage, then slam them to the ground left and right. Do you have how Hulk slaughters Loki in the first Avengers? There, like that. Beyond that, our fairies can create a ladder to use as we like, which proves to be a great escape route at many times.
Technically Rodland is bursting with Japan, colorful graphics, everything is very cute and sugary, even the music is more than appropriate to maintain the Japanese anime atmosphere.
Excellent playability, very simple but not at all easy, the difficulty level in fact gradually increases until it becomes quite arduous, there will be many situations where everything will seem much less cuddly, trust me. Absolutely worth mentioning is the possibility of playing in doubles, the fun becomes epochal as well as the swearing, in fact with our wand we could turn on the (theoretically) allied fairy by slamming her here and there.
Rodland is a colorful treat, an irresistible bon bon bon that guarantees abundant fun mixed with extreme rage!

**BC KID**

Another conversion, this time not from Coin-Op but from consoles, PC Kid (with a "P") was in fact the mascot of the glorious PC Engine (wonderful machine).
The conversion is the work of the guys at Factor 5, authors of the three episodes of Turrican on the Amiga, so a guarantee of absolute quality. If we then add the music rearranged by Chris Huelsbeck, it is easy to understand why this conversion has been considered by many to be even better than the original in many respects.
In BC Kid we will impersonate Bonk, a small caveman equipped, however, with a huge head that will also be our only offensive weapon. Our task is to eliminate the evil dinosaurs that populate the different levels until we reach the final boss.
As mentioned, all we will have at our disposal to survive is the protagonist's big head, and I am not referring to gray matter. In fact, in the course of our adventure we will have to headbutt anything that moves and wants to hurt us, which personally has always given me some satisfaction.
Again, somewhat like Rodland, it is all wonderfully Japanese cartoony. All the characters that populate the world of BC Kid are aesthetically silly and adorable, the whole thing has always reminded me a bit of Doraemon as an aesthetic.
Graphically clean, colorful and without the slightest uncertainty, the absolutely spot-on music then is a treat to the ears.
The playability is great, everything is very simple as a platformer often needs to be but it is fun like few things. Bc Kid is one of those games that I call "with a smile," impossible not to play it smiling all the time, at least for people like yours truly who grew up on Japanese anime! An absolute must-have title for platform fans and beyond.

**MR NUTS: HOPPIN' MAD**

The story of Mr. Nutz on the Amiga is well known: given the now inexorable decline of the Amiga in 1993, the title originally planned for our machine of the heart was released for the Snes and Megadrive. It is a good platformer, nothing striking but more than decent. However, it was decided to release Mr Nutz also on the Amiga but it was not a trivial conversion but a completely new title, excluding the protagonist everything was redone from scratch and the game turned out to be completely different from its console counterparts. Not just different but better.



**MR NUTZ HOPPIN' MAD**

As Mr. Nutz, an anthropomorphic squirrel, we will be called upon to save the world from alien chickens (don't ask).
This great title is proof that to make a great game you don't always need to be original; on the contrary, in this case the guys at Kaiko (Apidya say anything?) draw heavily from the absolute platformer references of the time: Mario World and (mostly) Sonic.
Looking at the map through which to access the different levels, it is impossible not to think of Mario World, with the difference that in the Amiga title the map is explorable in search of objects and characters to talk to. Having started the first level instead, then moving on to the actual platforming, the mind immediately goes to Sonic. Both aesthetically and in terms of gameplay, it is evident how much the blue hedgehog was more than an inspiration. The pace is very high (not supersonic, but still fast) and even in the bonus levels that border on psychedelic, the inspiration is evident.
Mr Nutz is technically sumptuous, with graphics saturated with extremely vivid colors, perfect scrolling, all accompanied by excellent music and effects.
The level design is remarkable, the levels huge and all to explore, completing Mr. Nutz is by no means quick and easy, requiring dedication and time excellently spent.
A must-have title that proves that sometimes copying is not so bad.

**RUFF 'N' TUMBLE**

Title that can be considered the swan song of our beloved Amiga, and what a song.
Released in 1994, by which time Commodore's fate was largely sealed and 32-bit consoles were making their appearance, Ruff 'n' Tumble is an absolutely outstanding title.
As Ruff Rogers, a child who ended up on an alien planet, we will have to defeat the powerful tyrant on duty, as well as retrieve lost marbles.
Ruff 'n' Tumble makes adrenaline-fueled pacing its strong point, thanks to some excellent gameplay gimmicks. In fact, the protagonist will have to shoot anyone in front of him, collecting the aforementioned marbles that will allow us to exit the level. The marbles will not be the only things that we will have to collect, we will in fact find weapons and power ups that will make our mouths of fire highly devastating, it is in the operation of the weapons that we find a great idea: the weapons have infinite ammunition but a time limit, which will lead us to devastate the devastable in the shortest time possible to make the most

**Ruff'n'Tumble**

of the weapon provided.
Ruff 'n' Tumble is visually spectacular, the graphics are top-notch, and everything flows smoothly without the slightest slowdown, despite the fact that very often the screen will be chock-full of enemies and explosions. The music is equally excellent, as fast-paced and adrenaline-pumping as it needs to be, and the sound effects are perfect.
The level design then is textbook, the levels are huge and full of secrets, although the game imposes a certain frenzy, ample space is still given to exploration and search for secrets.
The four environments divided into 16 total levels are challenging, partly because of a difficulty that is not low but never unfair; every 4 levels we will face the ever-present bosses, which are huge and spattaculous.
Ruff 'n' Tumble is must-play title, it does not allow a moment's respite and your desire for destruction will be amply satisfied.
Released out of time, Ruff 'n' Tumble is an act of love to our dear Amiga and shows what Mother Commodore's machine was really capable of.

Dear friends and dear friends here we come to the end of this rich list that, given the quantity and level of the many platformers released on Amiga, could be definitely longer, in fact there are many titles that deserve to be remembered.
So the invitation is always the same, come visit us on our website and social channels and tell us your list.

Greetings to you all and AMIGA FOREVER!

NEW GAME

# NEONNOIR

**Year**: 2022/2023
**Editor**: Steamknight
**Genre**: Avventura grafica
**Platform**: Amiga
**Website**: https://steamknight.itch.io/neonnoir

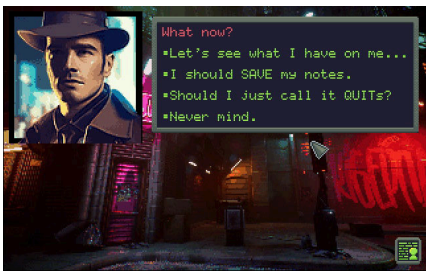
NEONNoir is an adventure game with cyberpunk themes made by Steamknight first appeared last year during the Amiga Blitz Basic Game Jam.

The game elicited mixed opinions because of impressive graphics, beautiful atmosphere but gameplay that seemed incomplete.
It now receives a new update that fixes some "dark" points and makes the story deeper.

The title puts us in the shoes of a private investigator searching for a young woman with a drug addiction problem.
The atmosphere here is that of Blade Runner or Shadowrun (the role-playing game).

What have they improved from the previous version? Certainly the game engine feels smoother. Dialog boxes that appeared confusing have been fixed, they fixed spelling errors and some incomplete sentences, the story is more understandable, and the game time flows more smoothly.

The title is basically a point-and-click graphic adventure with a minimal inventory system.
It is entirely playable with the mouse, and it is not necessary to click on every object and everything to advance



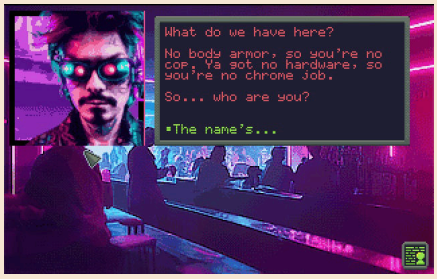
in the puzzles.
The game requires an A1200 with 2 Mb ram chip, 4 or more megs of Fast Ram, a hard disk with at least 5 mb of space.

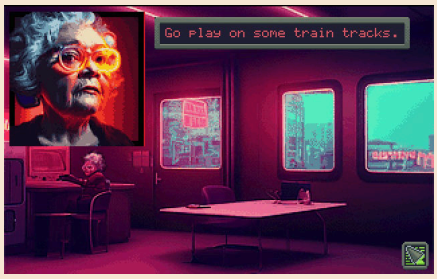
It works, of course, in emulation via the most popular emulators. A manual for proper installation has also been released on the developers' itch.io site.

The title was developed entirely in Blitz Basic and C++, and the source code is available on the same developers' page.
Technically it is certainly well done and with beautiful atmospheric music.
It is not a long title, nor is it complicated. In about two hours of play you will solve the case.

For those who get stuck in the game however, a walktrough has been included.
Definitely an interesting product for lovers of cyberpunk and noir games.

by **Carlo Nithaiah Del Mar Pirazzini**



## OUR FINAL SCORE

**» Gameplay 70%**
Point-and-click system and on-screen narration. Simple and effective.

**» Longevity 60%**
Not difficult and unlikely to replay.

NEW GAME

# LESTER

**Year**: 2023
**Editor/Developer**: KNIFEGRINDER
**Genre**: Platform/Action
**Platform**: Commodore 64
**Website**: https://knifegrinder.itch.io/lester

It is really a bad feeling when you find yourself alone, armed with only your little laser gun and an entire space station out of control.
This happened to poor android L. 3573-R, the first of a new generation of AI-independent guardians of the space station.
First and most unfortunate in that the ship's artificial intelligence went mad and decided to bring the nuclear cargo directly against the planet earth to destroy it.
So the task of L.3573-R is precisely to reach the nerve center of the ship, destroy the evil AI and save the world.
A simple task on paper but a complicated one as the ship teems with fierce enemy droids and deadly traps.
The title is inspired by modern metroidvanias released on MSX such as Ghost (from 2017) and older Konami productions such as The Maze of Galious.
To reach the ship's control center we will have to retrieve several access keys and disentangle platforms and opponents.
During the adventure it is possible to upgrade our armament and recover lost hit points through terminals. In order to buy energy and weapons we will have to collect coins that opponents and downed objects will release to us. The terminal also serves as a map of the ship and allows us to figure out where we are inside the ship.
At first it appears to be a simple game, but as we get deeper into the levels the difficulty increases, and it will be increasingly complicated to keep going without proper upgrading of one's arsenal.
A small rpg mechanic has been included. Each time you destroy an opposing robot, it will raise the experience bar, which, once maxed out, will allow us to have better health.
Lester is a well-made game. Solid, with precise and very responsive controls (perfect when played with a joystick) that work perfectly with one-button gameplay.
The graphics are detailed, clean, and well animated, reminding us of those of some Nintendo NES titles with special attention to detail (the backdrops are beautiful).
The sound compartment is beautiful with the SID churning out for us a beautiful soundtrack and fast-paced sound tracks perfect for keeping the action going at its best.
Nice little introductory sequence as well. Masterful work.
The title is compatible with pal and ntcs standards, and from our tests it runs correctly on both real hardware, emulation and The C64 family (mini and maxi).
Although there are 60 rooms to explore, it is not an extremely difficult title to complete. It requires commitment and some manual dexterity for the more difficult levels, but it is enjoyable and not at all frustrating.
The title is available for free on the developer's page on itch.io. However, I recommend that you leave a small tip for this work because it deserves it.
Really a good start for the Commodore 64 in this 2023.

**by Carlo Nithaiah Del Mar Pirazzini**

## OUR FINAL SCORE

**» Gameplay 95%**
Responsive controls, well-done level design. A great game

**» Longevity 90%**
It consists of 60 rooms and allows you to play for a long time. Not difficult nor impossible but fun.

NEW GAME

# LIFE OF MARS

**Year**: 2022

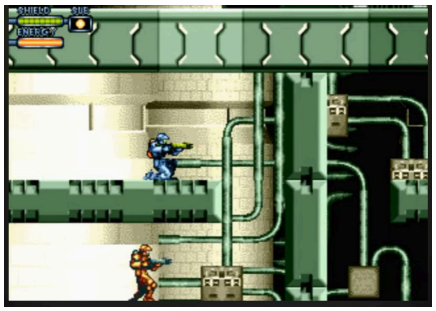
**Editor**: Kai Magazine

**Genre**: Action/Platform

**Platform**: Sega Megadrive

The group behind Life of Mars is Spain's Kai Magazine, former authors of the entertaining Metal Dragon and Life of Mars itself for MSX.
The person responsible for the project is Oscar Kenneth Albero, who first made the version for the MSX system and then converted it and stuck it into a cartridge for Sega Megadrive.
We are looking at a MetroidVania, that strange mix of platforms and action found in classic Metroid and Castelvania in which jumping and shooting is as important as exploring an open environment and discovering its secrets.
Life o f Mars takes us several years into the future where we will don the space suit of Sam, who is in charge of investigating a Martian colony that no longer shows signs of life. Once inside the Martian research center, we will discover that there is more "life" than you might think...but it's damned hostile!

The game is very reminiscent of Metroid because of the exploratory style, but I also saw a lot of Turrican in it because of the level design and type of enemies.
Nice animations but a little careless of the backgrounds, and the palette is perhaps too "off." The sound is a reproduction of that of the MSX 2 version, but it does not excel and

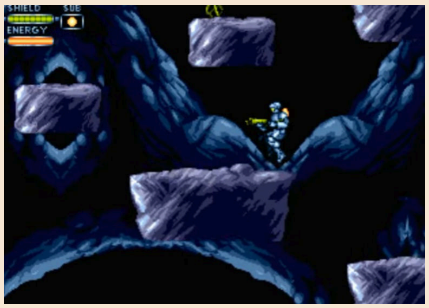


often "croaks."
Exploration is pleasant and engaging in the story but in some places the control system of our character is not the best and we will often find ourselves making "pixel perfect" jumps in order to reach some simple platforms.
I would have pushed harder in terms of exploration. It is perhaps more inclined toward the standard platform game than the classic MetroidVania.

The MSX 2 version had given me a few hours of play and the difficulty level was well calibrated, this Megadrive version suffers from the not "exceptional" controls and is frustrating in some places.

The title is available on Kai Magazine's official website at a price of 55 euros, perhaps a tad high for this type of game.
Suitable for lovers of this genre.

**by Giampaolo Moraschi**



## OUR FINAL SCORE

**» Gameplay 75%**
The control system is not precise and often the action becomes unenjoyable.

**» Longevity 65%**
Not very long... And 55 euros is a lot if you finish the title quickly.

NEW GAME

# BUFONADA

**Year**: 2022
**Editor**: Roolandoo
**Genre**: Platform/Puzzle
**Platform**: MSX/MSX2 – Spectrum
**Website**: https://www.msxdev.org/2022/09/06/msxdev22-16-bufonada/
**Website**: https://roolandoo.itch.io/bufonada

Roolando is the creator of that little gem S.O.L.O another title that participated in the 2021 MSXDev that we reviewed in issues 36 ita and 14 uk with good ratings.

He is a developer who knows his job well and churns out titles that are fun and well-curated.

This Bufonada is a medieval adventure set in the kingdom of Kimeria, a nation struck by a powerful curse that has cast it into eternal night.

Our hero is Jester the Jester, who finds himself with the onerous task of figuring out how to bring light back to his home world.

To do all this he makes use of his powerful spiritual animal-a goat! One often expects to have beautiful, powerful, and majestic spiritual avatars, and yet poor Jester is left with a bleating, rather ill-tempered little goat. In order to summon her, the jester will have to cast a small ball of energy et voila! The little goat will be at his full disposal.

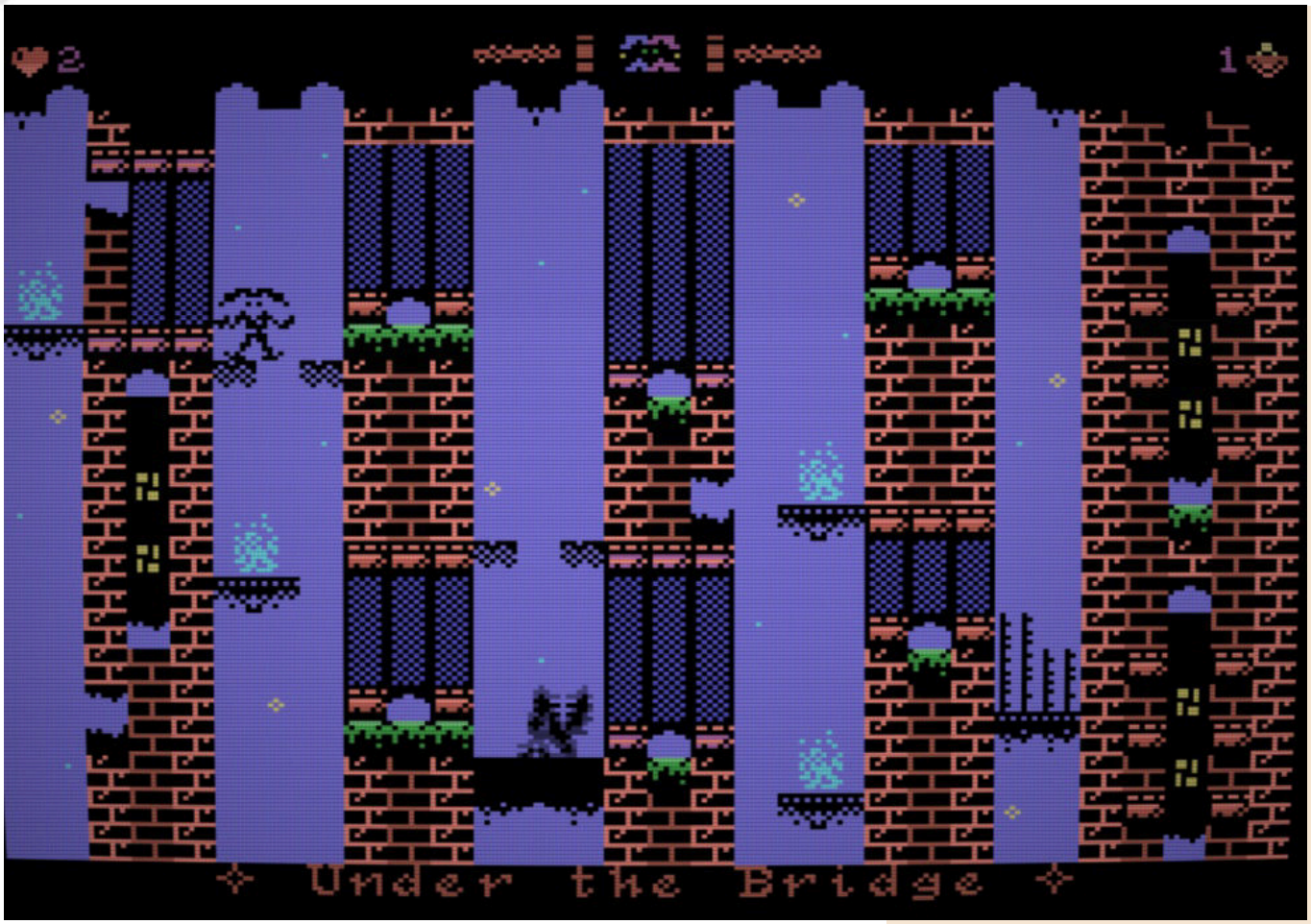
Bufonada is a classic fixed-screen platformer that brings both aspects of our protagonist into play. Jester can move, jump and use objects but cannot reach all platforms and needs to harness his goat animal spirit to do so. This involves the ability to switch during the game between one character and another. All with the press of a button. The quadruped will help Jester overcome certain points by acting as a trampoline, use its horns to break objects, and be useful for activating buttons.

A game system that is different and fun and totally functional. It may be off-putting at first but it works and glues you to the joystick.

You are an imbecile but I need you. We must stop the invasion. Take this magic orb and you can summon your spirit animal.
Midday
Jester's House

It is a brilliant title with perfect game design.

Graphically it is well realized and detailed also thanks to a "night" palette that creates atmosphere and beautiful animations (both MSX and ZX versions).




Atmospheric and rather sinister audio compartment that evoked some Arcturus pieces for me (who knows them?).
It is not an easy game and will often lead us to repeat the level several times. Partly because confused by the switch partly because some levels are diabolical however it is a great game that keeps us attached to the screen in both its editions.
I preferred the smoother MSX version but the ZX version is also a good game. Highly recommended.

by **Giampaolo Moraschi**

## OUR FINAL SCORE

### » Gameplay 90%
Careful game design that mixes action, character switches, and beautiful levels.

### » Longevity 90%
You will initially swear in several languages, but once you figure out how to best combine the two characters, you will look forward to completing it.

NEW GAME

# MAD STALKER
## FULL METAL FORTH

**Year**: 2022
**Editor/Developer**: Strictly Limited – Family Soft
**Genre**: Beat em up
**Platform**: Sega Megadrive
**Website**: https://www.strictlylimitedgames.com/collections/sega-mega-drive-genesis-partner-store

Mad Stalker: Full Metal Forth is a classic side-scrolling beat em up made does Family Soft/Fill in Café in 1994 for Sharp's X68000.
It was quite successful on Japanese soil and was later converted and released on FM Town and Pc Engine CD ROM.

In 1997 it also found a nice conversion for Playstation 1, and a few years ago the development team had begun a porting for the 16-bit SEGA as well. After a couple of years, here it is in our hands.
The title is reminiscent of Kung-Fu Master, and we take on the role of a police officer who takes control of the defense mecha known as the Hound Dog.
The task will be to rid Artemis City of crime and defeat another mecha called Rising Dog, similar to us but devoted to crime.

The game is linear in nature, the player will move from left to right controlling the mecha with the pad and eliminating the numerous opponents that will pop up on the screen. Task easy to write but difficult to put into practice as each opponent will require a different tactical approach.

What makes the difference is Hound Dog's level of combat readiness. Our robot features a fighting style worthy of the best Street Fighter II warriors, complete with special moves and hit combos.

There are 6 game areas to deal with, and at the end of each of these is the most classic of boss fights.

A title that presents nothing new compared to many other similar titles but that on paper and, joypad in hand, turns out to be a lot of fun.

It has always reminded me of two things. The first was the "Patlabor"-like setting (an anime detective series with robots instead of classic police officers) and two titles I particularly loved: Cybernator and Genesis Target Earth.

This Megadrive version suffers technically from the years. It is identical to the X68000 version (slightly slower), the robots are animated well while backdrops and colors are nothing impressive.

I recommend trying it out to see if you can like it or not. I found it fun and enjoyable. Definitely a game where you do not need to think totally but where you need to beat hard and beat hard.

Full Sufficiency.

by **Giampaolo Moraschi**

## OUR FINAL SCORE

### » Gameplay 80%

It can be enjoyed, and the controls are well calibrated. Fun combo moves and numerous attack types.

### » Longevity 70%

Do six levels seem like a lot or a few? It depends on how you approach the title. It is not eternal but it is enjoyable and you will often replay it in a relaxing break.

NEW GAME

# A PIG QUEST

**Year**: 2023
**Editor/Developer**: Protovision – Piggy 18
**Genre**: Platform/Adventure
**Platform**: Commodore 64
**Website**: https://piggy18.itch.io/a-pig-quest

Frank Further is a pig. He lives in Porkville, a peaceful, gilt-edged, laughing town where everyone is happy. Food, sunshine, puddles, apples and gladness are present in this pretty paradise.

But in the life of a pig, not everything is perfect--at least not everything until the day the welfare statues disappear, Lady Porchetta is kidnapped, and every inhabitant of the city is transformed into a lazy, motionless, and uncaring do-nothing. Little Frank goes on an adventure (despite himself) by accidentally ending up in a cave and throwing himself into the most important adventure of his life.

Thus opens one of this year's most anticipated titles for the Commodore 64, A Pig Quest is perhaps one of the examples of how you can still pull off something incredible on the 8-bit Commodore through passion, knowledge of the system and sense of level building.

It is a platformer reminiscent of Ghost 'n Goblins but not only that, it would have been nice but also too obvious. Then the Piggy 18 team slipped in elements from titles such as Zelda or Castelvania, transforming it into a title with nonlinear gameplay where exploration takes center stage and where you will need to be as careful as possible to unlock hidden keys, secret doors, picks, trapdoors, and traps.

But that's not enough. There are a ton of mentions of other titles both visually and in the game system.

Little Frank will be able to roam the 5 worlds not only armed with a precision knife and health undershirt, but by unearthing weapons and armor in the many chests present (which must be opened with a key). Each weapon has different characteristics (Pork's hammer is incredible) and the armors will allow for greater survival. The levels are never dull and have incredible variety of enemies and situations, and that's the beauty of

this title, you have to think before you jump and blast, and we like that.

There will also be hidden secret areas to discover and the narration that will often come our way to understand how to approach a given situation/monster.

This marvel was created by the collaboration of Antonio Savona on code, Mauricet on graphics and concept, and Aldo Chiummo and Gaetano Chiummo on music and sfx.

Chapeau guys! There is no history.

Detailed graphics. A riot of colorful pixel art with apt color choices. Perfectly animated and varied, even too much so. There is so much to see in A PIG QUEST, and the player will get lost exploring levels and dissecting their secrets just to want to get to the next world.

The SID in this game has been taken and taken to the max. The thirty-six different songs in so many settings (not only in the levels, but also in the game paintings) are made with such love that they move you. An impressive level of music.

There is everything and there is more. A calibrated difficulty system mixed with excellent joystick control. There are the really impressive end-of-world monsters that require determination and a dash of experience.

The levels are magically structured, carefully designed and thought out, thought out, loved--there is truly love in A PIG QUEST.

There is passion and knowledge.

As I write these words, the digital and boxed version has not yet been released (it is Feb. 8 and the title comes out on Valentine's Day), but I strongly recommend that you rush out and purchase it.

The title is available in .crt format in a digital version that is fully compatible in real hardware (both Pal and NTSC) and with emulation systems and the THE C64 (maxi and mini).

The box version will be in cartridge format.

It is worth every penny and you will not regret wanting to go down into the caves together with Frank.

by **Carlo Nithaiah Del Mar Pirazzini**

## OUR FINAL SCORE

**» Gameplay 95%**
Everything lines up perfectly. Balanced, fun levels and excellent controls.

**» Longevity 95%**
Only 5 worlds it seems few... But there are 200 screens to tackle, study and play.

NEW GAME

# AQUABYSS

**Year**: 2022
**Editor/Developer**: Aged Code
**Genre**: Action-adventure, Strategy
**Platform**: Amiga
**Website**: https://aquabyss.com/

A lavish and ambitious new Amiga title; priced not quite "homebrew" and with innovative mechanics (at least on the Amiga).

Aquabyss after years of development comes to market at Amiga 37 in Monchengladbach, Germany, and arouses curiosity-

It is a game for a retro machine but with demands of modern times. The minimum hardware to run the game is an Amiga 68000, 2 mb chip, 8 of Fast and 15 mb of free hard disk space.

But, because of the particularities of the game (such as visits to cities) at least a 68020 at 28Mhz or, even better a 68030 at 50 mhz or higher is recommended.

The game draws heavily on the network connection. It can be installed locally through an executable, but the rest will be downloaded from Aged Code's server. An unusual system on the Amiga. Each time the game activates and connects to the server it checks for updates. If present, it downloads and installs them automatically.

Another interesting feature is the list of results between players and the ability to interact with other players through an online game chat.

The game can also be played completely disconnected from the web. It is up to the player to decide whether to connect to the server or not. It is also possible to limit online play by downloading only updates.

The game setting is one dear to the worlds of Jules Verne and Steampunk themes.
Humans made large cities in the deep sea, similar to their surface versions. Within these cities an extensive trade

network has developed. The purpose of the game is to trade, exchange and administer the trade network and also uncover a small mystery, the disappearance of one of our relatives.

We will move among the seven seas with our trusty submarine. An amazing machine but one that constantly asks for upgrades, modifications, purchase of new fuel and maintenance.
Exploration of the game confronts us with numerous possibilities. The cities are many and are arranged in numerous underwater zones (not all of which can be reached because of the different depths), there are various possibilities for interaction, exchange and purchase, and it is possible to explore this fascinating adventure in all its parts.

Aquabyss is a game that requires patience, willingness to learn the mechanics and a lot of depth. It is not an immediate title, requiring reasoning and a high dose of learning.

I recommend following the story mode and reading everything that comes on the screen to learn all the mechanics and dialogues. Without it you will get stuck very soon.

If you're looking for action, forget it. It is a repeating title that sets in motion mechanics already seen in similar PC titles such as Sid Maier's Pirates for example. The gameplay is definitely not varied, and this may turn the nose up at those looking for something different.
Technically it is beautiful to look at and with a nice soundtrack.

It is not an inexpensive title (for the times and for Amiga), but it is a different experience that I would recommend and support.

by **Roberto Pirazzini**

## OUR FINAL SCORE

**» Gameplay 85%**
A Steampunk novel to play.

**» Longevity 75%**
It is essential to follow the text and story.
Gameplay may be repetitive.

NEW GAME

# MARIA RENARD'S REVENGE

**Year**: 2023
**Editor/Developer**: Z-team
**Genre**: Platform
**Platform**: Amiga 1200/CD32/ A500 mini
**Website**: https://z-team.itch.io/maria-renards-revenge

Maria Renard's Revenge is a title made by Z-team, a group that particularly likes Konami's Castelvania saga. A title developed with the help of the Scorpion Engine, a development engine that has been doing so well in the Amiga scene in recent years.

But who is Maria Renard? She is a character from the Konami saga. A young vampire hunter who boasts a distant kinship with the Belmont family/ clan.

She was only 12 years old when she first appeared in the series, but she was already endowed with a powerful knowledge of magic, familiar spirits and celestial beast magic.

The first title where he appears is Castelvania: Rondo of Blood where he will clash with Dracula himself.

This is a rather beloved character in the series who changes the way he approaches the many titles in which he appears.

In this Amiga fan game, our heroine must traverse 4 levels filled with monsters and traps and reach ... the evil that haunts her world.

The title was made from the graphic maps and sprites of Konami titles and presents pleasing, pixelated graphics. Everything is very colorful and pleasant, and the many animations are enjoyable as is the accompanying soundtrack.

The game is free to download from

## OUR FINAL SCORE

**» Gameplay 60%**
Some programming problems compromise the result and playability.

**» Longevity 60%**
From fans for fans of Castelvania.
The difficulty needs to be re-checked.

the developer's site and is available in several formats.
The version for A1200/CD32 (where 1Mb of Fast is enough), the version for A500 mini, and a Windows version that launches the rom in emulation mode.
Let's get to the bottom line.
It is nice but not dancing is said in this case. Graphically, as I said, pleasing but some little things are missing (which I sincerely hope will be corrected in future releases). There are numerous graphical glitches that do not make some platforms well understood. The game system is not the smoothest and we often die without realizing what hit us. Add to this a strange balance of difficulty that sees the first level very "tenacious" and the later ones rather "smooth."
In short, it is a game that has been expected by the community but has obvious flaws.
Too bad.

by **Carlo Nithaiah Del Mar Pirazzini**

NEW GAME

# S.A.K.

**Year**: 2023
**Editore**: SAK team/Future Disk
**Genre**: Action
**Platform**: MSX2
**Website**: http://futuredisk.nl/fd47.php

Green Beret (Rush n Attack) was one of the most beloved and at the same time most hated arcade titles ever.

Released in 1985 by Konami it was remembered for its "cold war" style setting and the absolute dependence the player had on having to use a knife to survive enemies.

A decidedly difficult but addictive title. Everyone wanted to see what was in the next picture.

This S.A.K for MSX 2 is a conversion of the classic game, indeed I would say a homage of the Konami title.
Four levels of play is an impressive

degree of challenge, all accompanied by smooth, colorful graphics and functional sound.
The game is still the same, and facing the "terrible Soviets" is not a piece of cake.

A title suitable for Green Beret fans and hardcore gamers looking for a challenge worthy of Hercules.

by **Giampaolo Moraschi**

## OUR FINAL SCORE

**» Gameplay 70%**
It is played well with the pad/ joystick and the title runs quite smoothly.

**» Longevity 65%**
Like the original title it will make you swear quite a bit...

# MYSTIC PILLARS

**Year**: 2008
**Developer**: Sivak Games
**Genre**: Puzzle Game
**Platform**: Nintendo NES
**Website**: https://www.retrousb.com/product_info.php?cPath=23&products_id=55

Inspired by the game Columns and the legendary Puyo Puyo, Mystic Pillars marks the third title on NES by Sivak Games.

The aim is to form with the tiles falling from above a row, column or diagonal of 3 or more matching tiles. In this way we will delete those tiles and all the tiles above will fall.

The further we go in the title, the more the game will increase level of difficulty and speed. Every so often special tiles will be released that will eliminate all tiles of the color where they rest.

Five levels, 3 music tracks and 8 customizable backgrounds are available.

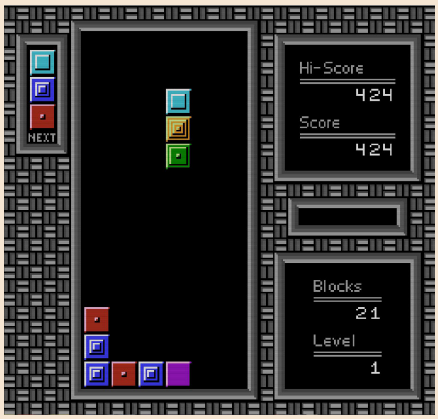
Mystic Pillars was supposed to be released in 2008, but was postponed due to numerous issues in sourcing materials to make the cartridges.

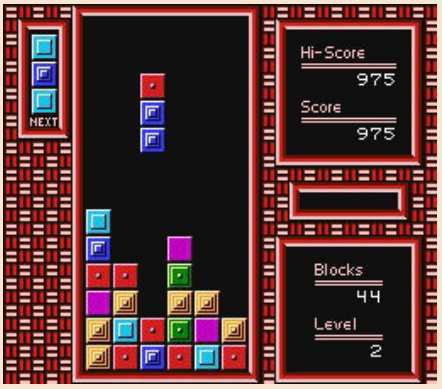
It is a simple and likeable title that certainly does not shine in originality but is enjoyable to play and super simple to learn.

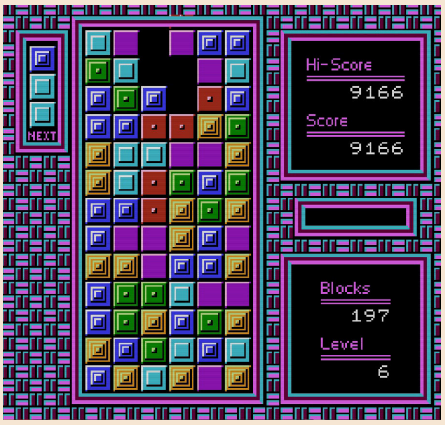
The difficulty level is well balanced, and this makes it enjoyable.

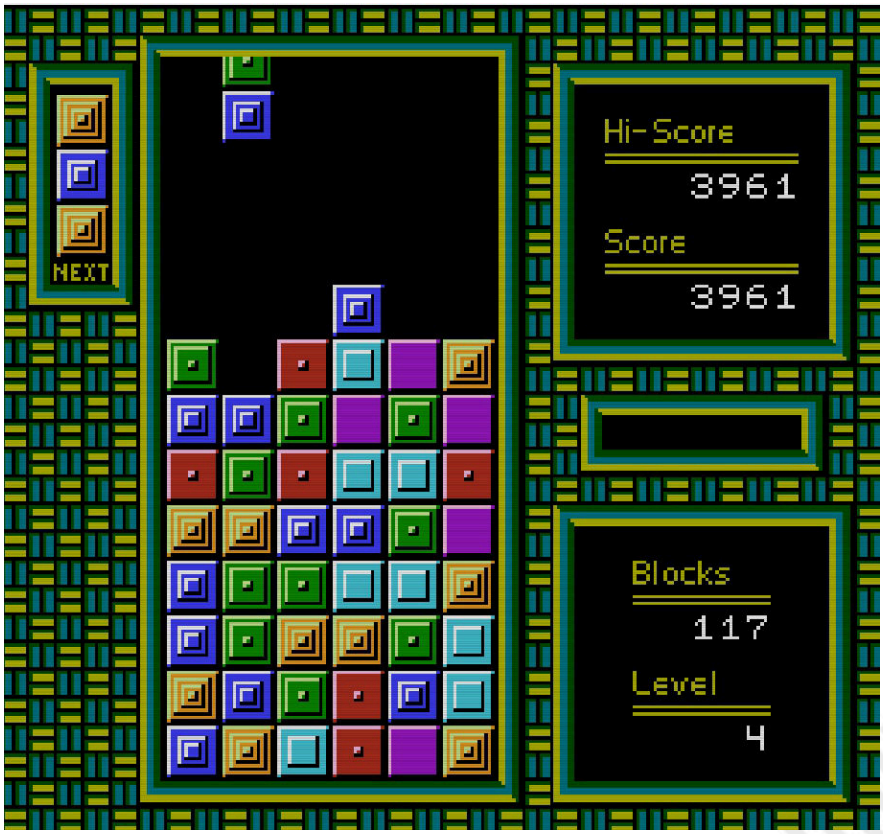
There are certainly superior titles, but Mystic Pillars deserves full sufficiency.

by **Giampaolo Moraschi**




## OUR FINAL SCORE

**» Gameplay 65%**
Nothing new, but it works well. Fair.

**» Longevity 70%**
Well-balanced difficulty.

# THE FIREMEN

**Year**: 1994

**Editor**: Human Entertaintment

**Genre**: Action

**Platform**: Super Nintendo

Year 2010, a well-known chemical company is holding its annual Christmas party on the premises.
A terrible fire breaks out in the kitchen, and before long, the entire building is on fire fueled by the many chemicals inside.
It's time to get in the game! The team of firefighters Pete (us), Danny, Max, Walter and Winona renowned for their bravery and skills throw themselves into the adventure to save the unfortunate employees and their families.
Of course worse Christmas Day just couldn't be had could it?
Fireman is a title that Uncle Nith gave me to try, and I must say I had a great time.

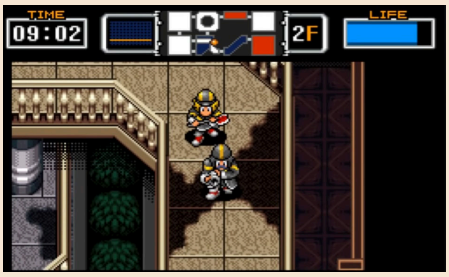
The most enjoyable thing about The Firemen is just how quickly the action-or rather, the flames-comes at us as we go from room to room, putting them all out. In essence, the game plays out like a race and gun meets bullet hell hybrid, with the fire hose as the only defense we have against being burned to a crisp.

We can use the hose in a couple of different ways: pointing it forward to shoot a jet in the direction we are facing to extinguish flames from a distance or pointing it toward the ground to deal with fast-moving fires along the floor. There are many original ways in which hell can threaten us as well.
There are rapidly spreading flames,

## OUR FINAL SCORE

**» Gameplay 80%**
Interesting game mechanics and nice levels to explore.

**» Longevity 70%**
Six challenging levels but in the end you won't replay it that often.



flames exploding from pipes, and even projectile fireballs that will test our reflexes. As if that were not enough, there are also thunderous explosions to deal with, the floor itself collapsing, and out-of-control service robots--and these are just some of the early levels! The Firemen presents only six levels in total. Things are reasonably manageable in terms of difficulty, but the later levels require a lot of precision and agility, with the fires spreading at an ever-increasing rate and the resulting damage becoming increasingly lethal. Fortunately, the controls are responsive enough to ensure that it really depends on our skills. The pace of play in The Firemen is often furious, but addictive, which is key to making this game an enjoyable experience.

Completing the game's six levels is no mean feat; the title's replayability once finished is rather limited. At the end of each level, we are given a score that details the amount of fire we put out for the particular floor you are on. So for completionists, there is a challenge to be met. But going back into the game again reveals the same fire outbreaks and level patterns. This is a completely pious (and nerdy) thought on my part, but random procedural fire generation could have been a great way to add some replayability. Hell, even a real level editor with levels saved on SRAM would have been great, but I'm sure it went well beyond Human's intentions with the game.

Nevertheless, The Firemen packs enough elements for it to be worth playing. The American action movie-style narrative, plus the fast-paced soundtrack that accompanies it, keeps things compelling, and the lighthearted anime-style graphics keep it from taking things too seriously.

That's it, I'm going back to putting out fires and hoses... the chemistry exam can wait.

by **Ingrid Poggiali**

NEW GAME

# MINKY

**Year**: 2022
**Editor/Developer**: Matze1887
**Genre**: Platform
**Platform**: Amiga
**Website**: https://matze1887.itch.io/minky

Minky is a new platformer for Amiga released in late 2022. It is an enjoyable and fun title that sees us in the role of the little monkey Minky who must pass 30 levels and retrieve all the coins present.

A simple, straightforward game that belongs to the "old school" of platform games.

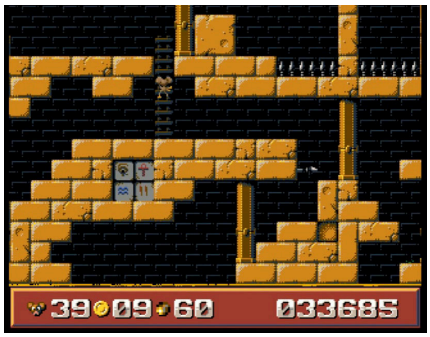
This is an enjoyable title to play and fun, challenging enough in the end but never to become impossible or frustrating.

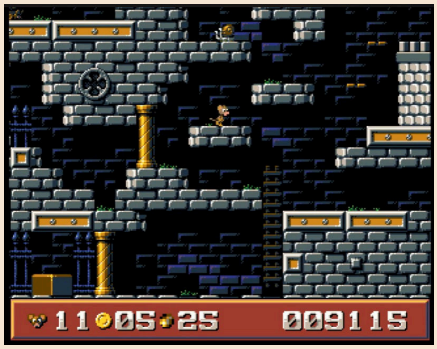
The password system to allow you to continue is done well and the levels are quite varied and enjoyable.

It reminded me of the Night Owl Design style of games, the sound is well done and is very enjoyable to listen to with some very pleasant melodies.

The only flaw I found is in the handling of jumps. Not a few times will be when you will be cursing as dockworkers to reach the most extreme tip of a platform.

Graphically it is very pleasing in its color choices and animations, a really well-made product.




You can play Minky in all possible ways: emulation, A500 mini, and on real hardware (game requirements are minimal, 1 mb of Ram).

It may not be the most original title ever released on the Amiga, but it is a good platformer.

by **Giampaolo Moraschi**

MINKY
INSTRUCTIONS
YOUR AIM IS TO COLLECT ALL COINS THAT ARE LOCATED THROUGHOUT THE LEVELS. EACH STAGE CONSISTS OF AN EXIT THAT WILL OPEN IF NO MORE COINS ARE LEFT. ENEMIES CAN BE KILLED WITH A JUMP ON TOP OF THEM OR BY THROWING A BALL!
SPECIAL POINTS CAN BE ACHIEVED!
- 200 POINTS -
NO LIFE LOST DURING A LEVEL.
- 500 POINTS -
FOR EACH COMPLETED LEVEL.
GRAPHICS (TILESETS):
SAUL CROSS

## OUR FINAL SCORE

**» Gameplay 70%**
Nice levels but the "pixel perfect" jumping system is lethal.

**» Longevity 85%**
30 levels is a lot to deal with, and the title can be happily played.

# SHADOW WARRIORS NINJA GAIDEN

**Year**: 1991

**Editor**: Tecmo

**Genre**: Beat em up

**Platform**: Game Boy

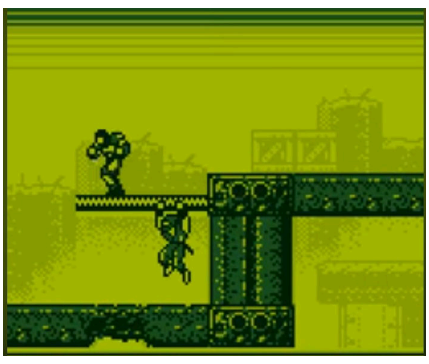
Ninja's Skyscraper Fight/Ninja Ryūkenden GB Matenrō Kessen/ Shadow Warriors Ninja Gaiden/Ninja Gaiden Shadow... Wow! A thousand names one game!

Hi! How are you doing? I hope well... I'm back with an all-Chinese treat to show you, the beautiful Shadow Warriors! Little appreciated by fans of our Ninja series, in my opinion it is a really crazy game if we put into account that it is on Gb!

Our protagonist Ryu Hayabusa will have to face quite a few, but he holds on for 5 stages ending with a boss.

The year was 1991 and, in Japan, the Tecmo team came out and said, "But if we made a Ninja Gaiden for GB would you like it?" So much they did and so much they said that, on September 13 of that year, the project arrived in stores. Ninja Gaiden fans generally snubbed it.

but that's no reason not to try it, since it rightfully enters the category of "Title you don't expect, " in a good way of course.



The reason Shadow Warriors is not particularly popular is related to the fact that it looks like a modified version of "Shadow of the Ninja, " a (well-known?) game by Natsume.

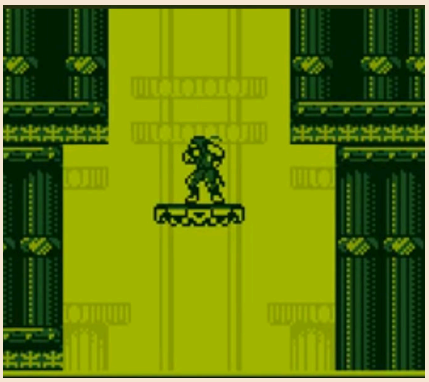
I personally strongly wanted to have it because it has really crazy graphics, with well-defined sprites and remarkable fluidity, which is uncommon for a GB game. Our character moves with great agility through the levels and is a pleasure to control. There is no shortage of various power ups, including the ninja rope "Ninchaku," which functions similar to the beloved mechanical arm from "Bionic Commando." We will not use Shuriken, but we will have a "katana smash everything ". Depending on the upgrades collected, we will also be able to make limited use of the "Fire Art Wheel" that will clear the screen of enemies.

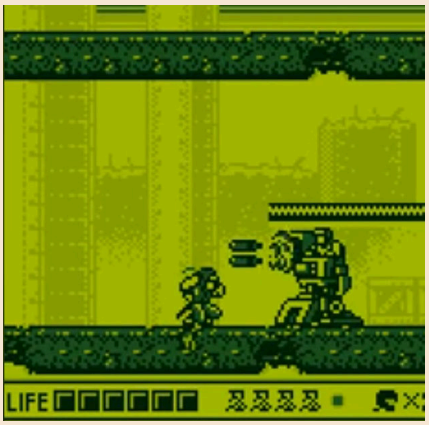
We will go through five pleasant stages facing Ninjas, Robots, Flying Robots, Laser Beams to avoid, tearing everything apart with our sword. One of the few flaws are the bosses, which are really easy and changeable (like Madonna in her heyday).

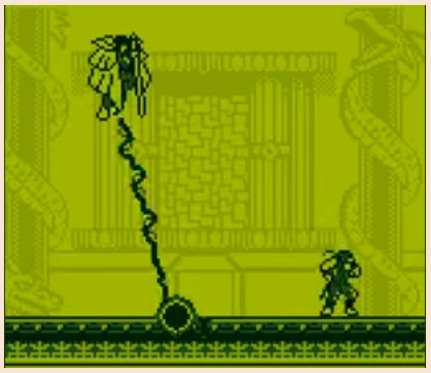
These will often be chameleon-like and discreetly unlikable: want an example? The final boss who starts as Dracula and after a while becomes a Mazinga Z-like robot!

Technically enjoyable and with good playability.

by **Barbara "Morgana" Murgida**





## OUR FINAL SCORE

**» Gameplay 75%**
Good level design and lots of stuff to destroy and beat up.

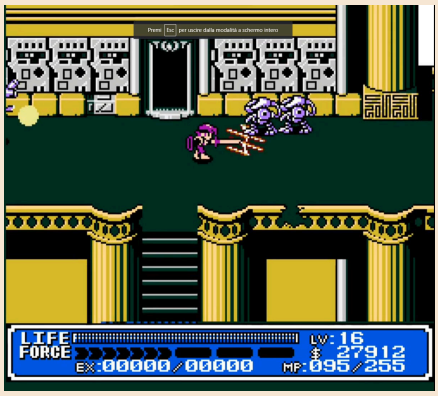
**» Longevity 75%**
Not a simple title but very enjoyable.

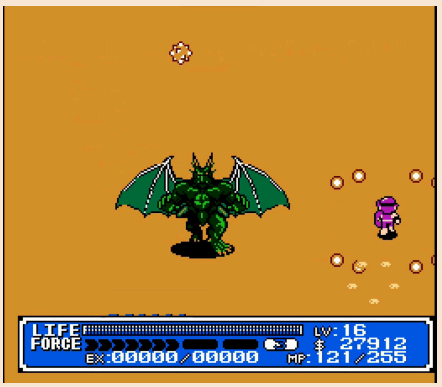
# CRYSTALIS

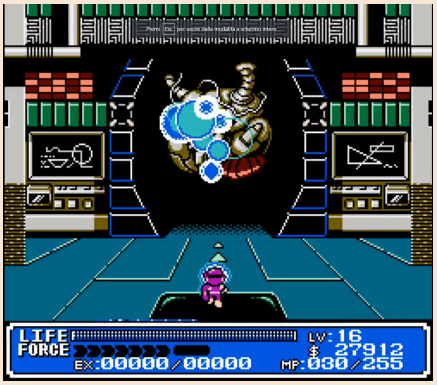
**Year**: 1990
**Editor/Developer**: SNK – Digital Eclipse
**Genre**: Action RPG
**Platform**: Nintendo NES/Game Boy Color

Crystalis is a 1990 action rpg produced by SNK for the NES.
The game begins with a young wizard awakening from a cryogenic sleep in which he had been imprisoned during the Great War and before the terrible nuclear apocalypse unleashed on the world in 1997.
He remembers nothing about himself or about who he was previously but gets involved in a great adventure that will lead him to save the world.
The title is made like many classics of the period, with a top-down view. The pad controls movement through the directional cross, the action button is used to fight with a sword while the secondary button is used to be able to use a magic power or an item summoned from the inventory.
It is possible to equip our hero with various armor, magic items and shields. Defeating monsters and solving quests allows you to earn experience points and increase our characters' level (Hp, defense, attack, etc.) accordingly. The main means of defeating monsters is through the use of four elemental swords scattered throughout the game (the fifth sword called Crystalis, is obtained only in the final world). These weapons can strike either with normal slashes or with attacks enhanced by the elementals contained within them. This elemental nature of theirs provides a necessary and effective means of dealing with certain situations.
Upon its release, the title was praised for its advanced graphical rendering that pushed the NES to its limits. We are talking about the console's end-of-life period, and the developers knew very well how to work on Nintendo's 8-bit. It is reminiscent of the style of Nausicaa in the Valley of the Wind. Very colorful, animated well and quite varied that is perfectly accompanied by an impressive and engaging soundtrack. Without a doubt the best soundtrack ever made on a Nintendo 8 Bit game.
Even now on a scenic level it is a rewarding title but has several problems in terms of gameplay.
The action is often repetitive, different enemies are all dealt with in the same way by continuously and quickly pressing the button.
Problem of repetitiveness in confrontations aside, the title is undoubtedly a quality product with a deep and well-written storyline.
Perhaps one of the best ARPGs for the NES. Worth rediscovering and playing.

by **Carlo Nithaiah Del Mar Pirazzini**




## OUR FINAL SCORE

**» Gameplay 85%**
Deep and engaging storyline.

**» Longevity 75%**
Repetitiveness of fights does not help and bores.

# JAMES BOND 007 LIVE AND LET DIE

**Year**: 1988

**Editor**: Domark

**Genre**: Azione

**Platform**: Commodore 64

We can already feel spring just around the corner, the days are tending to get longer, and unfortunately or fortunately, Milan has been without snow this year. I hope this new year is off to a very good start for everyone after two terrible years of restrictions fighting with an invisible but fearsome enemy. Anyway dear fans of 007 and Buggy Boy I have the pleasure to tell you about Live and Let Die, one of the many games based on the film saga of the secret agent James Bond 007 that since the 60's gives us entertainment, intrigue,twists and turns thanks to the charm of the very good cast, in addition of course to numerous titles for computers and consoles. This Domark-branded game developed by Elite Systems is a combination of racing and action game starring a rear-view speedboat piloted by us and James Bond. The style of gameplay is very reminiscent of Buggy Boy which almost everyone knows like the back of their hand by now with many similarities such as jumping logs, ramps, tunnels and a few minor differences: the race takes place on water instead of on a track, the lack of checkpoints and instead of a countdown timer, a fuel tank that slowly empties like our bank account when we are in Courmayeur in the middle of the season. We will be able to fire infinite ammunition plus three missiles at more armored enemies and even knock down walls.

There are four courses and each will keep you busy for a good ten minutes or so. We begin with the practice course in which there are only rocks and red and black targets. The former we will have to hit them with ordinary firing, while the latter with missiles; the aim of the course is rather simple, which is to hit all the targets. The second and third levels are also drills set in Antarctica and the Sahara Desert, but this time there are many more head-scratchers such as enemy speedboats, mines, planes dropping bombs, moving turrets, etc. The fourth and final track is the actual mission: in New Orleans we will have to get all the way to Mr. Big's refinery (an antagonist also present in the film) to blow it up. If you already want to know how, I'll just tell you right now, don't waste the missiles because even though a helicopter will occasionally pass by and provide them, many times it will instead shoot its own way without even giving you a nod.

Playability-wise it is nothing much once you get the hang of it. Separate discussion deserves the shooting: many times I found it difficult to hit enemies especially diagonally preferring rather to avoid them. The soundtrack follows the movie soundtrack but only in the opening screen, in the rest of the game you will only hear the roar of the speedboat and the sound of gunfire.

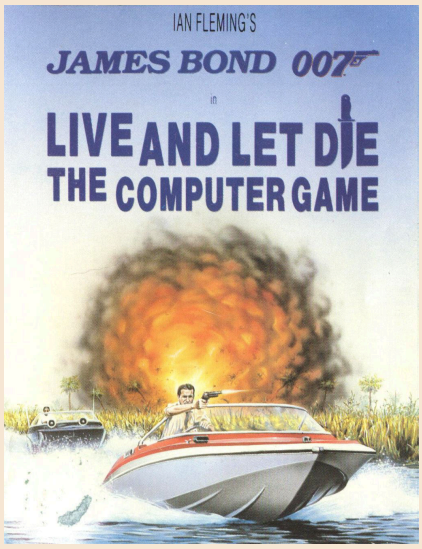
The only way to end a game is to run out of gas. And here I feel obliged to give you a piece of advice; always go at top speed and do not neglect the fuel canisters you will find along the way, marked by these colors: red for minimum amount, green for average and gray for almost full.

In terms of longevity four paths might seem few, but you will see that it will be enough given the difficulty and the time it takes to get to the refinery.

There are no saves, continues, credits in the game because it does not follow a linear story. Once you finish a route, the game will end and you will decide the order in which to play: all routes in a row or one in violta.

Happy early spring and enjoy this title of a legendary saga!

**by Daniele Brahimi**



## OUR FINAL SCORE

**» Gameplay 80%**
Slightly erratic shooting but still enjoyable.

**» Longevity 75%**
Four levels, not many but pretty tough!

There was a time a few years ago when interacting with friends and colleagues reminiscing about the days of our youth, I began to wonder if lingering in certain memories was nothing more than an unhealthy regurgitation of our own childhood. All these adults talking about cartoons, singing their theme songs, or computer platforms that had almost disappeared, was both touching and a little sad.

Probably with a hint of arrogance, I used to tell myself that in my case there was something more beyond appearances, but without really knowing what. Actually something more was really there: the desire not only to remember but also to do, to make up for "lost" time and take nostalgia to a "new level", transforming a simple poignant feeling into a powerful and modern course of action. It was certainly not an attempt to feel better than others (by no means am I that type of person), but only to give substance to something that was in danger of being lost.
As many readers will know by now, my video game "career" began in 1982 with a friend's VIC-20 and the first game ever played was "The Count", a text adventure by Scott Adams. Since then, text adventures have remained in my heart and, although I actually played very few of them (some were so difficult: English was still an "alien language" to me and guides or solutions could not be easily found), they are still one of my favorite genres.

A few years later I bought my first computer, a Commodore 128, a machine as beautiful as it was misunderstood and therefore snubbed. The available software was scarce and the games basically were'n there (unless you switched to 64 mode, but that's another story). I was forced to write all the programs I needed myself, which was certainly very educational, but sometimes the results were poor. In particular, the few text adventures I wrote were little more than a technical demo, not at all presentable outside my small bedroom.

So, what does this have to do with the opening remarks?
Well, for a start, the main point is that one should never give up and lose hope; second, there is always time to move to another stage; third, nostalgia and infantilism are only in the eye of the beholder. As Bruce Wayne said, "It is my actions that qualify me."

Last month, I took the plunge and entered PunyJam#3, a competition among IF (Interactive Fiction) enthusiasts that aims to introduce the "Inform" language (created by Infocom to develop all its games, starting with "Zork") to the coding community. Although I did not know Inform, I jumped into the fray, learned its rudiments, and wrote a complete text adventure. As "basic" and all in all linear as it may be considered, it is fully functional and modern tools allowed me to compile it to play on my favorite 8-bits, starting with my trusty Commodore 128. A few days ago I turned 53 and a once unattainable piece of software is now absolutely real and ready to be shared with the community. So much for nostalgia and childhood; the future has never been brighter or more "retro" than this.

**Gianluca Girelli**


# Disclaimer

RetroMagazine World as an aperiodic magazine entirely ad-free is a non-profit project and falls off any commercial circuit. All the published material is produced by the respective authors and published thanks to their authorization.

RetroMagazine World is licensed under the terms of: Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) https://creativecommons.org/licenses/by-nc-sa/4.0/

This is a human-readable summary of (and not a substitute for) the license. You are free to:

**Share** — copy and redistribute the material in any medium or format

**Adapt** — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms. Under the following terms:

**Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

**NonCommercial** — You may not use the material for commercial purposes.

**ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

**No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



RetroMagazine World-Eng
*Year 4 - Issue 20 - OCTOBER 2023*

**Editor in Chief**
*Francesco Fiorentini*
**Editing/Deputy Manager**
*David La Monaca*
**Publishing Manager**
*Marco Pistorio*
**Website Manager/Social Media Manager**
*Giorgio Balestrieri/Carlo N. Del Mar Pirazzini*